University of BRISTOL

Department of Computer Science

A Computer Poker Player

Dominic James Harvey

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

January 2017 | CSMSC-17

Declaration:

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Dominic James Harvey, January 2017

Executive Summary

This work reviewed existing artificial intelligence research into poker and extended it to fraud detection in online poker for the first time, with a focus on automated poker players (‘bots’). Successful methods for detecting fraud were highlighted, developed, and tested.

In terms of the categories given for the project, this work did the following:

Type I (Software/hardware development):

- Java was used to write a user-friendly analysis tool that players can use in conjunction with commercial software to review game data for signs of fraud.

Type II (Investigatory/research):

- The behaviour of poker players, site operators, and bot developers is shaped by public perception of the threat posed by bots. Poker players were questioned extensively to give a public account of this perception.

- Previous fraud investigations were examined to highlight successful strategies for detecting fraud in online poker.

Type III (Theoretical):

- A theoretical account was developed of differences between human agents and machine agents in strategic settings. This was used to suggest modified poker variants that are uniquely difficult for machine agents for sites to use as part of their offering.

- The literature on opponent modelling was reviewed and extended to questions of fraud detection. It was shown that a revised frequentist method adapted from applied poker analysis can outperform traditional frequentist approaches.

- This method was applied in a discussion of methods for quantifying the difference between poker strategies. It was conjectured that previous Euclidean distance measures would be outperformed by a modified Euclidean measure that accounts for important features of poker strategy. Game data showing suspected fraud was tested to compare the effectiveness of distance measures under varied conditions.

Acknowledgements

My sincere gratitude goes to:

- Ian Holyer, my supervisor, for his reassuring stoicism and guidance
- Oliver Ray and Tim Kovacs, for their support in navigating the course
- Other university faculty members who offered their expertise, especially Ayalvadi Ganesh and Oliver Johnson
- The poker researchers who were generous with their time and professional insight, especially Sam Ganzfried, Michael Johanson, and Eric Jackson
- The poker hobbyists who have been immersed in this issue for years and were eager to help, especially Roger Mainfield and Casper Greve

Contents

1. Introduction

"After humanity spent thousands of years improving our tactics, computers tell us that humans are completely wrong... I would go so far as to say not a single human has touched the edge of the truth" - Ke Jie, world Go champion

"I can't help but ask, one day many years later, when you find your previous awareness, cognition, and choices are all wrong, will you keep going along the wrong path or reject yourself?" - Gu Li, 9 dan Go professional

Since the birth of artificial intelligence, researchers have used games as settings for exploring new topics. Games present clearly defined rules and goals in which success can be usefully measured and a game can often be found that shares the salient features of the target problem.

The properties that make poker a deep and interesting game are found in many difficult and important problems. Poker players act with imperfect information (opponents' cards and future community cards are unknown), cannot always access hidden information later (opponents' cards are only sometimes revealed), and must account for the game's stochastic nature (random cards dealt in future rounds can change the value of players' hands). Unlike in many games a poker player is not trying to satisfy a binary win/loss condition but seeks to maximize their winnings. Poker is played against dynamic agents who can adjust their strategies between hands, making opponent modelling a challenge. The concept of poker covers a large family of games making it easy to find one with a suitable structure for addressing a problem. Popular forms of poker offer human experts to test against and a well-developed body of theory to use a frame of reference.

At the turn of the century Schaeffer, who was later to solve checkers, wrote that, "Successfully achieving high computer performance in a non-trivial game can be a stepping stone toward solving more challenging real-world problems" [1]. Since then, techniques developed in efforts to solve poker have been applied to problems in a wide variety of domains including civil engineering, security [2], and medicine [3]. These efforts have also conquered new frontiers in artificial intelligence: Computer Poker Research Group's Cepheus program solving Limit Hold'em in 2015 was the first solution for a game played competitively by humans with imperfect information or stochastic elements. In his history of early research in the field, Billings remarked, "Somewhat surprisingly, the potential benefits of studying poker have been largely overlooked by computer scientists and game researchers" [4]. Those benefits are now readily apparent.

The idea of computers "beating humans at their own game" attracts significant public and commercial interest. The matches between Garry Kasparov and Deep Blue raised the public profile of artificial intelligence research and spurred the development of more sophisticated game-playing programs. AlphaGo's recent million-dollar victory over Lee Sedol and subsequent domination of online Go (the title quotes come from two opponents in AlphaGo's 60-match online winning streak) marked a breakthrough in a game that had resisted attempts by machines to compete at a high level. The solution of Limit Hold'em was a milestone not just for researchers but for the poker community. The week before this thesis was submitted saw the Cepheus team announce major developments in No Limit Hold'em just before the start of the Second Man vs. Machine No-Limit Texas Hold'em Competition between top human players and an advanced poker program from other researchers.

The average poker player engages with artificial intelligence in a very different context. Online poker is a multi-billion dollar industry supporting millions of recreational players and tens of thousands of professionals. An increasingly relevant issue for online poker is the rise of automated players or 'bots' operating on poker sites in explicit violation of the Terms of Service and often winning large amounts of money from opponents. Players have a direct financial interest in seeing this problem fixed and site operators have an interest in appeasing them; nearly half of the players surveyed for this work said they have stopped or would stop playing on a site with a known bot issue.

Despite the impressive developments in artificial intelligence research into poker, very little of it is relevant to this problem. For that research poker is a means to an end: a specific domain for finding techniques that are not domain-specific. Bots are developed in academic research for conditions that do not represent poker as played by most humans. These bots are often built for relatively simple poker games that are not played competitively as part of efforts to find solutions for the game; finding these solutions is not realistic for popular variants like No Limit Hold'em or Pot Limit Omaha. The theoretical guarantees offered by these solutions only apply for two-player zero sum games but most online poker is multi-player. Many bots are designed to follow a static, equilibrium strategy while online players want to exploit weaknesses in their opponents. In contrast, bots in use in online poker act under the opposite conditions as well as unique constraints such as avoiding detection. Despite this, they are still winning and winning enough to cause concern.

1a. Distribution and explanation of work

This paper aims to give a formal treatment of this problem from a range of perspectives.

Chapter 2 gives the relevant background information needed to understand the work.

Chapter 3 covers common human misunderstandings about poker, AI, and poker AI, and takes a detailed look at how humans understand poker strategies and how this mode of understanding is uniquely different from the way in which machines operate. This difference is then exploited to find ways of modifying poker games to make them less susceptible to bot use while retaining their appeal to humans. This chapter also contains a comprehensive survey of public opinion regarding the threat bots pose to online poker.

Chapter 4 gives an overview of different approaches to opponent modelling and explores the idea of teams of agents. It notes common problems encountered by these methods, advocating and formalizing an approach from applied poker analysis.

Chapter 5 analyzes previous investigations into bot use and other types of fraud, giving empirical evidence in favour of this selective statistical approach.

Chapter 6 deals with the theoretical problem of measuring the difference between two strategies, which is essential to tackling fraud in a scientific fashion. It argues that the Mahalanobis distance better fulfils these requirements than typical Euclidean distance measures and tests this using a data set containing suspected bots.

Chapter 7 explains the PokerParser utility that was developed for this project and used for the experiment in the previous chapter.

Chapter 8 summarizes the work done in the project, evaluates it, and offers suggestions for future work.

2. Background

2.1 Terminology

Activities such as poker or chess are colloquially understood as 'games'. In game theory, the term refers to a model of agents' strategic choices and their outcomes under a set of conditions. Difficult real-world problems are often more tractable if modelled as games and analyzed under this framework.

The first step of this analysis is choosing an appropriate model. Sequential games such as poker are usually modelled as 'extensive-form' games. Formally, these involve:

- A finite set of *players*, N={1, ..., N}. In games with stochastic events such as flipping coins or drawing cards, a 'chance player' is included to represent these.

- A finite *game tree*, with the set H of possible action histories represented by nodes in the tree. Terminal nodes $Z \subseteq H$ represent payoffs to players. For non-terminal nodes, a *player function* P assigns the node a player; for $h \in H$, $P(h)$ denotes the player to act at the node representing h. If, at each node, a player retains all information they held at previous nodes, the game is said to have *perfect recall*.

- A partition function grouping game states into *information sets* where the player to act cannot distinguish between different game states in the same information set. For imperfect information games it is common to group game states that are identical except for the information hidden to the agent (in poker, opponents' private cards). For large games it may be necessary to group states that differ in other ways for the purpose of constructing a smaller, abstract game. Finding a sensible and efficient method for this is essential; see [5] for discussions of this in the context of poker.

- A utility function that determines the utilities assigned to players at terminal nodes: for $z \in Z$, $u_i(z)$ gives the utility for player i at z. If $N = \{1, 2\}$ and $\forall z, u_i(z) + u_2(z) = 0$, the game is *zero-sum*.

A full, general definition can be found in Osborne and Rubinstein [6].

For a player i, a *strategy* $\sigma_i \in \Sigma_i$ $_\mathrm{i}$ assigns a probability distribution over possible actions at each information set. σ_{-i} denotes the strategies for all players other than i; together, these make up the *strategy profile* σ. The *best response* or *nemesis* for i is the strategy that maximizes their utility against the rest of the strategy profile. A strategy's *exploitability* is the difference between its utility when facing its nemesis and the highest utility a strategy can achieve in the game. Crucially, a *Nash equilibrium* is found when each player's strategy is simultaneously a best response to their opponents' strategies, such that no player can improve their expected payoff by choosing another strategy; a strategy belonging to a Nash equilibrium has zero exploitability. When finding an exact Nash equilibrium is unrealistic, an approximation is sought instead. An ϵ-Nash equilibrium occurs when the increase in utility to a player from changing their strategy is capped at ϵ; for sufficiently large ϵ, this includes a broad range of strategies. In games challenging enough to appeal to researchers, computing the exploitability of a strategy is itself a difficult task.

This paper makes frequent references to *‘solving [a] game’*, which is typically used to mean finding Nash equilibrium strategies. In the poker community, a strategy that is a member of a set of Nash equilibrium strategies is referred to as 'game-theory optimal' ('GTO'). However, this term is used loosely and inaccurately, even by top players [7], indicating that the idea is widely misunderstood.

Throughout this work, “bot” is used as a general term for poker-playing programs; this includes those that require human supervision, but it is assumed that a supervisor is not interfering with the bot’s play. ‘Commercial’ is used to distinguish bots developed for research purposes from those used on online poker sites for personal profit, though many of these bots are developed by hobbyists and not a primary source of income or available for public sale or inspection. I use “poker variant” to refer to a specific set of rules that govern how poker is played - No Limit Hold’em, Pot Limit Omaha Hi/Lo - and ‘[poker] game’ to mean instances of those variants, such as a ‘full-ring £1/£2 No Limit Hold’em game’.

For hand examples the standard poker terminology is used: ‘Ah’ refers to the ace of hearts, 98o refers to an offsuit combination of a 9 and an 8, etc. The examples are mostly illustrative, so the reader should not infer anything from the choice of hands or terms used for these.

2.2 Poker

The rules of most poker variants can be summarized briefly:

- Each player is dealt a number of private cards. Forced bets such as ‘antes’ or 'blinds' are extracted from players to initialize the 'pot', which is awarded to the eventual winner of the hand. This begins the first betting round.

- In each betting round, if no bet has been made, a player may 'check' to pass the action to the next player or make a bet. Once a bet is made, each other player in turn must choose to match this opening bet, raise the bet if allowed, or fold. This continues until all players have folded or contributed equally to the pot. In the first betting round, the blinds act as these initial bets.

- Between betting rounds, the game state changes. Usually, some new information is revealed in the form of a new community card, with a burn card removed from the deck without being shown; in some variants, players may exchange their private cards for new cards.

- When all but one player has folded, that player wins the pot. Otherwise, at the end of the final betting round, players' private cards and the community cards are used to determine the winner based on predetermined hand rankings.

Most play occurs under one of two formats. In cash games, players buy in for their chosen amount and can top up their chip count as desired, with the blinds remaining constant. Success in this format is measured by a player's net winnings in a session (analytically, by a player's expected winnings); a player's strength is judged by their win rate over a large sample size. In tournaments, players receive a fixed number of chips and are knocked out of the tournament when all of these are lost; the blinds change over time to accelerate this process. Testing for automated poker agents is often conducted in a modified tournament format; see the Annual Computer Poker Competition for an example [8].

Early work in game theory was illustrated using Kuhn Poker, a simple abstract game [9]. Similar ‘toy’ games are still popular in AI research and poker analysis [10] as they provide an accessible testing ground for new theories. Academic efforts to develop competitive poker programs over the past twenty years have focused on Limit Hold'em. This was the most common poker variant until the mid-2000s, offering the benefits described in the introduction. It is both complex enough to present interesting challenges and simple enough that overcoming these challenges is realistic. Solving a game of this size was a major achievement for research in this domain. However, despite the effort needed to solve it, Limit Hold'em is much simpler than other popular games as it restricts players to a fixed bet size. Lifting this restriction substantially increases the size of the game tree requiring the use of abstraction techniques to make the problem tractable [11]. The recent focus has been on No Limit Hold'em, the most popular variant and a continuing challenge for researchers. The next most popular poker variant online, Pot Limit Omaha, has received surprisingly little attention, but the techniques developed to tackle Hold'em variants and the concepts discussed here extend to other poker variants as well as other domains. The format of the game is important too: most research is done in two-player or 'heads-up' games, whereas most online play involves six to nine players at the same table. This difference has important theoretical and practical implications.

3. Human understanding and implementation of strategies

Poker analysis software plays an important role in online poker. Game theory programs like GTORangeBuilder and PioSOLVER allow players to isolate and dissect specific situations. Data analysis tools such as Hold'em Manager and PokerTracker sift through large databases of hand histories to give users a better understanding of their own and their opponents' play. Heads-up displays or 'HUDs' that allow statistical profiling of opponents in real time are ubiquitous in competitive games. This demand creates lucrative opportunities for developers who can present strategically useful information in a format that users can understand.

Meanwhile, recent advances in poker AI have prompted questions about the implications for poker at large. Limit Hold'em is solved but what does that mean for Limit Hold'em games online or in a casino? Many successful players have a strong mathematical intuition but few have a formal education in computer science. Poker AI research may hold some academic interest but their focus is on finding actionable advice to improve their play.

Poker site operators hoping to unmask bots in their games are keen to find patterns of behaviour that distinguish humans from bots. When bot developers prioritize avoiding detection and having their bots appear human, operators want to know how this is attempted and how successful it might be.

Each case raises questions about how human players understand and implement poker strategies. Despite its fundamental importance to poker and commercial importance to software developers and site operators, this problem has received little coverage in academic poker literature. This section surveys the state of research in this area and addresses the problem from a variety of perspectives.

When researchers seek solutions for large games, these are not intended to be adapted for human use. To the extent that these solutions are useful for humans it is because the process of finding them aids the development of techniques useful in other domains; there is no incentive for researchers to consider the poker-specific implications of their work. The logistical challenges of finding these solutions are formidable. For any game large enough to be interesting for competitive play, the computational effort needed to find and store a solution greatly outstrips the capacity of human agents [3]. Reducing this to a form that humans can understand would require a highly lossy abstraction and the solution produced will not be faithful to the original. More generally, most poker AI research aims to test the limits of what can be done using current techniques. Accessibility by humans is a significant constraint that is only considered when necessary.

This dynamic is explored for the first time in a recent paper by Ganzfried and Yusuf [12]. They apply machine learning techniques to a game built to resemble No Limit Hold'em endgames, suggesting that the results can be output as simple decision trees that human players can easily understand and memorize. Their method generalizes to large, imperfect-information games and should be useful for poker variants that humans play competitively. However, there is no experimental data showing how simple these rules have to be for humans to memorize and follow them competently. We would expect a tradeoff between increased simplicity and the coarseness of the abstraction used for these rules; 'abstraction pathologies' encountered in attempts to solve large games via abstraction suggest that, even if an agent opts for a more fine-grained abstraction at the cost of harder comprehension, there is no guarantee that the resulting strategy will be more robust [13]. Progress in this area will require constructing a sound model of this tradeoff.

There is also no general account of which methods of representing a strategy are best suited for human use. Insights from other domains may be relevant here. Specifically, noting the obstacles humans face in implementing even basic strategies, it would be useful to have an account of how likely and severe human error is in this context. If the expected impact of human error is high, we may prefer a strategy that is more exploitable but less susceptible to misunderstanding; if two strategies have about the same exploitability, we may choose the one for which we expect the impact of human error to have less variance. Successful players must be self-aware enough to recognize their own limitations and modify their play to fit these. For instance, the training tool PokerSnowie recommends raising preflop from early position in No Limit Hold'em with more suited Ax hands and fewer suited connectors like JdTd than the average player does. Brokos' explanation is that using suited Ax (a stronger starting hand) for this purpose, despite being a more theoretically sound play, results in a top-heavy raising range: an opponent can apply pressure on a flop that does not play well with high cards, knowing that our hand is unlikely to have improved. Defending against this aggression with an optimal frequency is easy for machines but harder for humans; defending too much or too little is highly and easily exploitable. Swapping parts of the bottom of the range for more low hands like suited connectors sacrifices some best-case equity in return for making the strategy easier for a human to implement in a balanced way against the likely exploitative tendencies of other humans [14].

This topic merits more attention from poker analysts as well as researchers. A large number of poker books, guides, and instructional videos are published every year [15], often commanding high prices. Many current and former professional players use poker coaching as a primary source of income [16]. These endeavours require assumptions about how different types of player learn and implement poker concepts. A rigorous study of poker pedagogy would help coaches and analysts to produce better content and allow consumers to make more informed choices.

3.1 Man-machine differences

A priori, there are reasons to expect observable differences between bots and humans even when bots try to imitate humans under ideal conditions. Suppose there is a known equilibrium strategy for a toy poker game. A human agent wishes to follow this strategy, which is basic enough that this is possible. At the same time, a machine agent follows this strategy in another instance of the game. It is not certain that the two agents will behave in the same way. A human will suffer from fatigue or become frustrated in a way that affects their play: responses to the survey in Section suggest that opponents are expected to show signs of this over a long session, and failure to do so is a strong indication that a player is a bot. Research shows that the emotional impact of bad luck or mistakes can have a strong negative impact on a player's performance [17]. They may be unable to follow the strategy properly: a strategy dictates that the agent perform actions with assigned frequencies, and human agents estimate these imperfectly. Recognizing this, the amateur participant in the highest stakes poker game of all time used external mechanisms to regulate his behaviour when making probabilistic decisions [18]. Even in this scenario in which identical or near-identical play is most likely - two players following the same, static strategy, with no incentive or intention for agents to deviate - it cannot be guaranteed.

When these conditions are relaxed, the problem becomes harder. These human errors introduce imbalances into the agent's play that an exploitative opponent will want to target; in turn, this creates opportunities for counter-exploitation. Despite initially following the same strategy, agents may deviate from it and try to punish others' deviation in different ways. A small change to one player's strategy creates compounding incentives for every agent to change their strategy resulting in a large overall difference. This is even clearer when players are actively trying to exploit opponents. As discussed in Section 4, opponent modelling remains a difficult problem for computer agents with sophisticated modelling techniques and many observations of the opponent's play. Human agents trying to model opponents and adjust their own play accordingly in real time can only do so haphazardly and will not resemble machine agents attempting the same task. Bot developers are forced to engage with this problem if they want to match human play as almost all human players try to exploit their opponents, to varying degrees of success. This becomes a bigger problem as the game becomes larger; most poker variants played competitively are so large that this issue is severe and unavoidable.

3.2 Finding man-machine differences

A poker site may have millions of registered players and tens of thousands active players at any one time [19] and high-volume players can log millions of hands in a year. Site operators have an omniscient view of an account's gameplay but need to know how to use this information to identify irregular behaviour. Conversely, individual players have to assess the likelihood that a suspicious opponent is a bot based on very limited information. Both parties seek an account of which pieces of information to look at and what to look for.

This account starts with decision points that a player encounters rarely. To the extent that players can approximate optimal play in a situation, this is largely possible because that situation occurs often enough for a savvy player to develop an intuitive understanding for it. These situations also receive the most attention in poker theory and analysis. For instance, the decision of which hands to call or 3-bet with when facing a preflop raise on the button is common enough that having a slightly better approach to it has a large impact on a player's overall winrate, so it makes sense for players to spend their limited time and resources focusing on spots like this. We expect the play of bots and skilled human players to be most similar here. In contrast, human play is weaker and more erratic in rare situations, where players lack experience and have less educational material to guide their play. A bot developed by an iterative method cannot, and does not need to, prioritize computational effort on a specific 'type' of decision and will be unusually balanced and consistent in rare situations. Bots developed with reference to examples of human play will have fewer such examples to work from.

This approach extends to multiplayer hands. Adding the possibility for more players creates a new family of situations that a player can encounter, decreasing the frequency of any specific situation and growing the game tree. Pots with many players are rare, offering the detection opportunities described above. Additionally, a hand with five players on the flop is often heads-up by the river; playing the entire hand as if it were multiplayer or heads-up is a mistake that, as before, humans will avoid differently from bots. Considering multiplayer games helps to narrow the scope of the problem: the theoretical guarantees relied on in finding equilibrium strategies for two-player games do not apply to multiplayer games and finding these strategies becomes much more complex. Despite this, bots are successful in 6-max and full-ring games on online sites (see Section 5) and their perceived popularity causes widespread concern. This suggests that the bots encountered online were likely to have been developed via other methods, allowing us to focus on looking for signs of these.

These classes of hands are examples of a more general point: situations that human players find difficult are a good place to look for irregular play. This underscores the importance of building a convincing account of poker pedagogy, if only for bot detection purposes.

3.3 Surveying perception of the bot threat

Public perception of the threat posed by bots informs the behaviour of all agents. Players decide which sites and games to play in based on their perception of how likely they are to encounter bots (see Section 3). Site operators trying to reassure players have to estimate how serious this threat is perceived to be and use the tools at their disposal to work out how serious it actually is. Bot developers wishing to avoid alerting human opponents engage in a trade-off between maximizing their expected payoff in a game and minimizing their deviation from ‘normal’ play. If most opponents are aware of the bot threat and paranoid about the safety of their usual games, this tilts the decision in favour of trying to appear normal; if the scale of the threat is underestimated, there is more room to maximize profit.

This dynamic is complicated by information asymmetries between agents. Players have no way of knowing how sophisticated and prevalent bots are at any given time. Bot developers are understandably reluctant to discuss their methods and individual developers do not know how advanced their competitors are. Site operators spend time and resources detecting bots but can only make educated guesses about the state of bot technology from a distorted sample: they cannot know about bots that successfully avoid detection. Site operators collectively know more than players and developers about the state of bot detection methods but this information is necessarily private. Most work in poker security uses proprietary software or is covered by non-disclosure agreements [20]. No one agent has a full understanding of the situation or an incentive to reveal how much they know, making research difficult.

The one area where productive work can be done is assessing the state of public understanding of poker AI. To this end, a survey was used to gauge the current perception of the threat posed by bots to online poker. This was conducted in two rounds: the first round was targeted at professional players, with 18 respondents. A slightly modified survey was used in the second round for a wider audience, with 29 respondents.

Questions were left open-ended instead of using a discrete scale. This makes preference analysis harder ("19% of players said they were very concerned about X, 26% said they were somewhat concerned...") but we were interested in detailed answers and didn't want respondents to shape their answers around the value they chose on a scale. Additionally, there is no clear consensus on the best way to construct this scale for a given problem [21].

Respondents gave their agreement that they would remain anonymous but that excerpts from their answers could be quoted.

1. Which games/stakes have you played regularly online in the past year? On which sites?

We hoped that a variety of sites and games would be feature in the answers to make them relevant to and representative of a wide range of players. We also expected to see issues mentioned more often by players on certain sites. In Round 1, this question did not ask about sites; this was corrected for Round 2 when we recognized the benefits of these details.

Round 1 respondents were split between NLHE and PLO with a slight lean to NLHE. Most respondents played MTTs: some exclusively high-stakes, but others covering a wide range with an average buy-in of $30-60. Some also played cash games, mostly at .50/1 or 1/2.

Round 2 respondents were concentrated on PokerStars, the largest international site, and played primarily low-stakes cash (.25/.50 or .50/1) with a handful preferring low buy-in tournaments. Note that, because of the complicated legal status of poker in the United States, the American player base is fragmented with sites like PokerStars only operational in certain states and most players spread across a number of US-only sites; this was reflected in the answers.

There is a probable selection effect here as players inclined to take a survey on poker bots are likely to play on sites where bots are seen as a threat; more generally, players who do not see a problem are unlikely to go to the effort of answering a survey, though some respondents did just that and didn't answer the later, specific questions.

2. How common and competitive do you believe bots are in those games? What effect are they having on players' winrates?

In both rounds respondents were evenly divided on how common bots are. There was a very strong correlation between believing bots are common and believing they are competitive: only two players who thought bots were common were unconcerned about their impact on the game. Multiple answers pointed out that, even when the bots themselves are not winning overall or aren't especially skilled, their presence still hurts others in the game; as one noted, "The bad ones are still beating a portion of the pool, but are also forcing the rest of the pool to play hands vs an additional opponent that they have little/no edge on, causing them to churn through extra rake and hands before they get involved with players they have an edge on".

The perceived impact of bots varied greatly between sites, stakes, and formats. Smaller sites were commonly named as easy targets for bots while most who play primarily on PokerStars said they had not recognized bots in their games. Round 1 respondents were more likely to be concerned about bots despite the added difficulty in building a bot that can win against more competitive players. PLO players were less concerned than NLHE players, while the respondents who play mixed games or uncommon games like PLO8 were confident they had faced no bots. Tournament players were less concerned than cash game players; several noted the difficulties the tournament structure poses for bots.

3. Have you decided not to play on a site because it has a reputation for being popular with bots? If not, would you consider this when deciding if you should play on a site?

This question was added to Round 2 to see how concerns about bots affected players' behaviour.

44% said the perceived popularity of bots had determined their choice of sites and games. Of the rest, some struck a defeatist tone arguing that there was no point as most sites had them anyway; others said it would affect their choice but there were no other convenient options. This suggests that, if for no other reason, sites have an incentive to establish a good reputation on this issue; this is predictably easier for larger sites with a higher volume of play (so any one instance of bot use will affect a lower proportion of total players) and more resources to devote to security. Several players said they played exclusively on PokerStars for this reason.

4. How much, and in what ways, do you believe bots have improved in the past few years?

Every Round 1 respondent who reported playing against bots believed they have substantially improved in recent years. Round 2 respondents were more divided - though one said they had improved enough that they “quit poker recently after playing professionally for nearly a half decade” - but it is unclear whether this is because professionals are more adept at noticing these improvements or because those improvements are necessary to keep up in tougher games.

Few specific improvements were mentioned but most replies spoke generally about a more balanced playstyle and fewer obvious mistakes. Some worried that newer bots were better at real-time exploitation.

5. What have you observed about the playstyles of suspected bots? How does this differ from the average player?

Round 1 respondents were split on whether bots were aiming for a balanced strategy or trying to exploit population tendencies; some distinguished between the two and said they had noticed both types of bot in different contexts.

Round 2 respondents observed a variety of differences from average players. The common theme was that bots were very aggressive and tended to be unusually skilled in handling uncommon situations, particularly on the river; this coincides with the analysis of common mistakes by human players given in Section 3.

6. What behaviours (gameplay related, as opposed to timing/chat activity etc) would cause you to suspect an account is a bot? How do these differ from the 'red flags' for other suspicious activity (such as collusion/multiaccounting)?

Interestingly, despite the wording, many Round 1 respondents chose to answer about factors like timing and chat activity. This may be because the previous question about bots’ playstyles was taken as a general question about strategy and it was assumed we would want to know about other behaviour. To avoid this apparent ambiguity, and because these answers were intriguing, there were clearer, separate questions for strategic behaviour and ‘player’ behaviour in Round 2.

For player behaviour, most said they had not noticed anything or did not think there were clear differences but those who did gave interesting answers. On timing: “The well programmed bots calibrate bet timing to appear human, which is not difficult because of the wide range of acceptable timing for human decision making. A red flag of a bot, though, is when they take non-human lines very fast. Usually it requires at least some thought to deviate from the norm". This extends the argument given in this work about the importance of identifying strategic difficulties that are quintessentially human. Another echoed the pre-selected bet sizing example given for the discussion of distance measures: "Weird bet sizes (not using the standard hotkeys for 3/4 pot for instance, but using 66.67% instead)". Lack of tilt and long sessions starting at odd times (often the same time, for multiple accounts) were also mentioned.

For strategic behaviour, one player noted the lack of flexibility in 4-bet/5-bet situations mentioned by Greve in Section 5; more mentioned that rare corner-cases are likely sources of different play, affirming the intuition given throughout this work.

Few answered the follow-up question about other suspicious activity; those who did noted that any deviant behaviour would be observable in differences in gameplay, but were not more specific.

7. How confident are you in your understanding of game theory?

This was a useful lesson in survey design and human psychology. This question was intended to show how knowledge of game theory correlated with understanding of actual and possible strategic differences between bots and humans. In practice, almost all respondents said their knowledge was strong (or better) so this was not possible. The most likely and charitable explanation is that this knowledge is necessary to succeed in the games that the Round 1 survey was aimed at. This question was omitted for Round 2.

8. What else do you think is worth noting about the bot situation at present?

The respondents who thought bots were a problem were very sceptical about site operators' interest in addressing it. An American player commented, "Bots are being found even in regulated and licensed legal online card rooms not just offshore sites. This epidemic is growing rapidly and threatens to destroy whatever hope that remains to having a viable healthy liquid player pool for US players on any site ever". Another was more direct: " It's eventually going to be the downfall of online poker and I hope the sites do enough to prevent that".

3.3.1 Evaluation

This was a productive and revealing exercise but there is room for improvement. As shown by differences in some of the questions between the two rounds, the design of a survey can have a big impact on the types of responses; details are given in the discussion of each question. For the sake of this work, a better question about players' abstract understanding of AI, in general and pertaining to poker, would have been useful.

In designing a survey there is a trade-off between detail for the designer and convenience for the user. Respondents were already being generous with their time and were required to explain their answers over the course of a long survey. If the survey was more demanding, there was a risk that players would think it wasn't worth their time. It is possible that not having a discrete scale already turned away potential answers. The survey was publicized on the largest poker forum and other media such as Twitter thanks to favours from well-connected figures; despite this, the response rate was low. It is unclear if this was avoidable. The answers seem fairly representative but the low sample size means that more complicated and useful relationships - for the sake of argument, "PLO players have a better understanding of poker AI but are less concerned about bots" - cannot be reliably identified.

Ultimately, the only parties with the ability to conduct this survey on a large-scale are site operators themselves who, according to a frequent complaint, do not appear interested in the topic.

3.4 Human misunderstanding of AI

A player's perception of the bot threat is informed by their knowledge of poker AI. Despite the academic background of many players and the mathematical skill or talent needed to play at a high level, even successful players often display a poor understanding of the field [7].

There are several reasons for this. As described, little effort has been made to make academic research accessible or relevant to players. Players have not found this a hindrance to bot detection in practice: a player may develop a keen sense for detecting bots based on in-game observations without a theoretical understanding of poker AI or general poker theory. Few papers contain useful information for this task and it is hard for those who do not follow poker AI research to identify these. Applied poker AI research often takes the form of public investigations by players in response to a recent scandal (see Section 5); these affect sites' reputations and shape the perception of how common and successful bots are. Finally, the relevant information in the public domain is spread between a variety of sources. It is hard to find these and assess their reliability: a poker fan with no background knowledge reading a noisy forum thread on the latest bot scandal will find it tough to identify who and what they should trust.

Within this, addressing popular misconceptions about poker AI is the most effective way to improve public understanding of the field. One such belief is that bots can never overtake humans because they cannot capture the psychological element of the game: humans can read their opponents and know how to outplay them in a way that a bot cannot. Often, this belief is rooted in a misunderstanding of both poker and artificial intelligence. 'Reading opponents and knowing how to outplay them' is a rebranding of a simple statistical process - constructing a model from observations and using that model to inform future actions - that computers already perform. There is no theoretical or empirical reason to believe that humans are somehow better at this than computers, or that there is something ineffable about this process that only humans can perform.

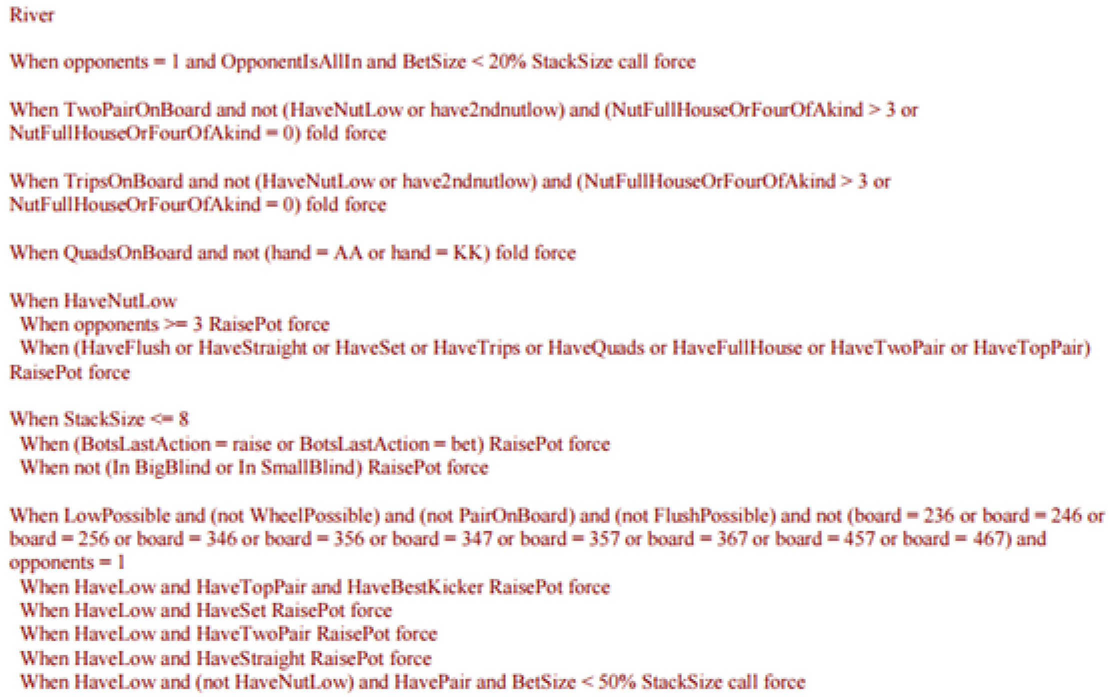
However, there is a kernel of a useful idea here. For the games in which commercial poker bots are found, finding a full GTO strategy - or even a rough approximation - is entirely unrealistic, especially with the limited resources available to most developers. Regular and average players in these games make significant mistakes that can be profitably exploited. Knowing this, most players try to target these mistakes. A player trying to follow a balanced strategy without engaging in active exploitation still has an interest in accurately profiling opponents: a set of strategies can have the same degree of exploitability but passively exploit (or receive "gifts" [22] from) different opponents. A player's expectations about the opponent will inform which strategy they ought to choose from that set. So all players have reasons to care about opponent modelling and many actively seek to pursue it; bots trying to appear human will have to follow this pattern to avoid suspicion. Since humans can only do this imperfectly, the errors they make create further opportunities for exploitation, and then counter-exploitation, and so on. This dynamic is hard to articulate - and still harder to have a bot replicate - but can be seen in observations of human play. This is a useful place to look for likely differences between humans and bots, not because humans are better at this but because they are worse.

Another common misconception is that the potential of poker AI is bounded by the limits of human understanding: a poker bot must be programmed to follow a strategy and this can only be as strong as the best strategy known to the programmer. This is clearly false for bots developed using an autonomous process: the team that solved Limit Hold'em by creating Cepheus using CFR+ (a variant of counterfactual regret minimization) was led by researchers with very little understanding or experience of poker [3]. When bots are manually programmed as below, expert knowledge is used to determine the shape of the strategy, imposing a ceiling on their performance. However, since a bot is not limited by the possibility of human error, it is better at implementing a given strategy. If an elite human player were monitored and their strategy reconstructed from examples of their play over a large sample size, this would look quite different from their intended strategy as understood by that player in real time. A strategy as conceived may be within the range of normal poker strategies but, if its implementation lacks the variation displayed by most humans in practice, the player will stand out. It may be a useful exercise to have players build bot profiles based on what they perceive their own strategy to be and test how their play differs from their bots', but this would be prohibitively difficult.

There is a larger question about what it means to 'understand' a strategy. Whether following a set of rules is enough to understand a process remains a contentious question in the philosophy of cognitive science [23]. The logistical demands of bot use provide an analogous example. Poker site operators use technical measures to detect bots such as mouse tracking or stacking table windows in an unpredictable way. To get around these, some bot users have humans perform the actions output by the bot on the site interface. The human 'player' takes actions with the frequency recommended by the strategy but we do not want to say the player 'understands' the strategy. Contrast this with the example given by Levermann [14] of players with no prior poker knowledge using poker training tools to become successful high-stakes players; their understanding derives from following the recommendations of the training tool rather than an abstract process, but they do seem to understand what they are doing.

This question also arises in bot development. While many bots are generated using methods that require no expert input, it is common for amateur bot developers to manually program bots: for example, a commercial developer offers a simple scripting language allowing users to create or customize player profiles (see a Poker Programming Language example in Figure 1). These bots seem more likely to emulate human play than those created via some self-contained process.

Programming a bot to follow a strategy requires a conceptual understanding deep enough to convert the strategy into another form (a linguistic representation). However, each step of this conversion is simple enough that the programmer's knowledge is not tested. Consider an example from the profile excerpted in Figure 1:



```
River

When opponents = 1 and OpponentIsAllIn and BetSize < 20% StackSize call force

When TwoPairOnBoard and not (HaveNutLow or have2ndnutlow) and (NutFullHouseOrFourOfAkind > 3 or
NutFullHouseOrFourOfAkind = 0) fold force

When TripsOnBoard and not (HaveNutLow or have2ndnutlow) and (NutFullHouseOrFourOfAkind > 3 or
NutFullHouseOrFourOfAkind = 0) fold force

When QuadsOnBoard and not (hand = AA or hand = KK) fold force

When HaveNutLow
  When opponents >= 3 RaisePot force
  When (HaveFlush or HaveStraight or HaveSet or HaveTrips or HaveQuads or HaveFullHouse or HaveTwoPair or HaveTopPair)
RaisePot force

When StackSize <= 8
  When (BotsLastAction = raise or BotsLastAction = bet) RaisePot force
  When not (In BigBlind or In SmallBlind) RaisePot force

When LowPossible and (not WheelPossible) and (not PairOnBoard) and (not FlushPossible) and not (board = 236 or board = 246 or
board = 256 or board = 346 or board = 356 or board = 347 or board = 357 or board = 367 or board = 457 or board = 467) and
opponents = 1
  When HaveLow and HaveTopPair and HaveBestKicker RaisePot force
  When HaveLow and HaveSet RaisePot force
  When HaveLow and HaveTwoPair RaisePot force
  When HaveLow and HaveStraight RaisePot force
  When HaveLow and (not HaveNutLow) and HavePair and BetSize < 50% StackSize call force
```

Figure 1 - Start of a river profile for a PPL bot

Pseudo-code: if I hold a nut low draw and a nut flush draw, make the maximum raise if I face a bet

PPL: When HaveNutLowDraw and HaveNutFlushDraw RaisePot force

To convert the pseudo-code into a correct PPL statement we need a basic knowledge of what that instruction means - what a nut flush draw is, what the maximum raise can be, and so on - but these are simply facts about the rules of the game. An expert player and a layperson following a rule sheet would have the same amount of relevant knowledge. The strategy is just the collection of these statements (and encoding the strategy is just repeating this process for each statement). Articulating the strategy takes a lot of work if the strategy is large but each step in the work remains simple.

To ‘understand’ the strategy, as the term is usually applied to humans, is to see it as ‘more than the sum of its parts’: drawing logical connections between parts of the strategy, seeing why actions are taken instead of the alternatives, and so on. To the extent that this distinguishes humans from bots, it explains the success of man-machine teams in other contexts. In ‘freestyle chess’ tournaments, which allow man-machine teams to participate, these teams outperform pure human and pure machine teams. The first large freestyle chess tournament was won by the team with the lowest individual Elo rankings, against teams featuring GMs or the strongest chess computers at the time, because they recognized and exploited the benefits of a man-machine partnership [24]. Previously, the widespread availability of chess engines had transformed chess education; commercial training and analysis software has had a similar impact in poker. It is unlikely that man-machine teams will have an edge over pure machine teams in poker, unless the human players have prior knowledge of how to exploit common design problems such as action mapping (see the footnote on p46), but top-level play between human experts increasingly resembles a contest between man-machine teams.

3.5 Modifying the game structure to vex bots

This ability to generalize existing knowledge and apply it to new situations can be framed more rigorously as finding shortcuts for computational work. Recall the basic structure of a poker hand:

```
= Private cards dealt (new information)
= Blinds posted (no new information)
= Betting round procedure

= For each postflop betting round:
== Burn a card (not shown, so no new information)
== Deal new card(s) (new information)
== Betting round procedure
```

Where betting round procedure:

```
= Until each player has acted and all have contributed the same amount or folded:
== For each player pi:
=== pi takes an action (if the agent is not pi, new information; if the agent is pi, decision point)
```

Adding new decisions or information causes an exponential increase in the size of the game tree, making an already difficult computational challenge much harder. Finding the size of the game tree is itself a hard task [11], so estimating the true extent of this change cannot be done here but, for a game this large already, any exponential increase requires a much greater computational burden. Human agents can adapt to these changes more easily. Solving the game at some point demands that the strategy have a plan for each possible information set. For instance, to solve the game on the turn is to know what to do on each possible river card: in normal Hold'em (HE), As, 2s, 3s...

Consider a modified Hold'em game, HE-burn, in which the burn card is shown instead of hidden. This greatly increases the range of possible information sets. For each river card, there is now a whole subtree of burn cards: As with the Ah burned, As with the 2h burned, and so on. The card removal effect from revealing the burn card (how this knowledge changes a player's strategy) is important: for instance, if the player holds an JT87 inside straight draw and a heart flush draw, any 9 or any heart as the burn card acts as a 'blocker' to a draw, decreasing the player's equity. This is particularly important in poker variants like Omaha where hands are closer in equity and players have more private cards so future community cards are more likely to expand the range of hand combinations a player can make. Grouping these information sets together for the sake of abstraction must be handled carefully and a relatively successful abstraction will still leave behind an enormous abstract game. Human agents will adjust to this change with varying degrees of effectiveness, but a competent human player can recognize and implement these adjustments despite being much weaker computationally than a bot. Following the discussion in Section 3, a good measure of a player's skill and understanding of poker is how quickly and effectively they adapt to such changes.

HE-burn shows that a simple modification to the rules has a remarkable impact on the size of the game and the size of a bot's comparative advantage over a human. HE-burn requires no additional actions from players and would be very simple to implement in live and online play. If we allow more complicated changes, we can be more creative. In HE-discard, each player is dealt three private cards and must discard one immediately; the discarded cards are revealed simultaneously before preflop betting starts. The hand then proceeds as a normal HE hand. Again, this presents an interesting challenge for human players and a tough challenge for bot developers. The timing of this information is crucial: if instead a card is revealed as soon as it is discarded, this has a massive effect on the importance of position and inflates the size of the game tree again preflop.

Finally, consider two variants proposed by the creator of PioSOLVER. In HE-fold1, whenever a player folds one of their private cards is chosen at random and revealed; in HE-fold2, the player chooses which card is revealed. In addition to the expected increase in the game size, this forces a reevaluation of some abstraction methods. When constructing a range of preflop starting hands it is common to group isomorphic hands: 'AK suited' includes AsKs, AhKh, AcKc, AdKd. The value of these hands becomes uneven depending on the revealed card: a diamond means that AdKd now has lower preflop equity. HE-fold2 is intriguing as the player occupies a unique position when choosing a card: their payoff from the choice is fixed at 0 as they are no longer active in the hand but the choice has implications for future hands if the game is understood as a repeated game. In contrast with HE-fold1, the reveal cannot be properly modelled as a chance event as the opponent's choice is not random; even if the opponent does not intend to play more hands and is effectively choosing at random, the agent has no way of knowing this. A bot developer is forced to engage in some type of opponent modelling to understand this action. As shown in Section 4, opponent modelling remains a frustrating challenge in poker AI.

The security implications are appealing. It creates extra obstacles for bot developers: a bot developed via an iterative process will have to start this again under new conditions. If the original process was computationally taxing, repeating it will strain these resources even further. For manually configured bots the programmer can perhaps draw on the human capacity for pattern-matching described above but reconfiguring the bot remains difficult and time-consuming. It also becomes much harder for bots to mimic human play. For a popular and long-running game like No Limit Hold'em players and bot developers have access to detailed theoretical analysis and large amounts of data as a frame of reference; most of this will not apply to the modified game. Bots cannot be designed to emulate humans if there is no clear idea of how humans do or should play. Less skilled players are likely to respond erratically to these changes and more skilled players will still take time to adjust properly, creating the 'time delay' problem described in the neural network discussion in Section 4: a bot that adjusts immediately and consistently will appear suspicious. Finally, whenever an agent must make a decision, different agents or types of agent will act in different ways; introducing more decision points into the game gives more chances for bots to distinguish themselves from humans.

This idea also has commercial appeal. New players can enjoy a fun twist to a familiar game while feeling less outmatched in terms of experience; skilled players can enjoy a fresh challenge that their knowledge lets them adapt to quickly; security-conscious players can play with less concern about bots. There is a worry that introducing too many variants creates a fragmented player pool but site operators can use their expertise to determine how many new games to offer and how these should look. Inspired by mixed games such as HORSE, in which hands of five different games are played in a rotation, one idea is a 'mixed Hold'em' game: a hand of No Limit Hold'em followed by a hand of HE-burn and so on. This would allow players only familiar with one variant of poker to enjoy the variety offered by a range of variants without learning a fundamentally different game. It would also force bot developers to create bots that are proficient at all variants in the rotation: a bot noticeably weaker or different in one variant will be noticed despite its play in other variants.

Regardless of how this is done, this analysis suggests that site operators consider adding modified games to their offering.

4. Opponent modelling

Poker AI research seeks to develop theories and techniques that apply in a wide range of domains. Many important problems, from physical and online security to military strategy to auctions, are modelled as competitive games that require a player to model an opponent's behaviour. In these settings an agent's best possible outcomes require exploiting an opponent's tendencies. In poker, analysis tools comb through hand histories to give descriptions of an opponent's strategy. Empirically, these tools have proven to be essential in competitive play. However, to use these tools is implicitly to take a stance on questions of opponent modelling. How much data is needed to have confidence in the model? When it is appropriate to act differently based on the model? How easily can a model of one opponent be generalized for common types of opponents? There are practical implications to ignoring these theoretical questions.

Opponent modelling presents distinct challenges for computer scientists. When problems are not well-defined or are hard for an agent to accurately characterize, prohibiting a strategy for the full game, we still want to use available information to inform our actions. A solution to a game does not necessarily give useful information about how to model opponents in the game. Furthermore, the difficulty of opponent modelling is not inherently related to the game's complexity. Rock-paper-scissors ('RPS') is a trivial game but opponent modelling remains tough. Billings et al. give details of a RPS computer tournament in which the goal is to win the most rounds; the obvious GTO strategy fails in this structure because it cannot earn above its fixed payoff, giving agents an incentive to pursue exploitation. Participants found that, "Contrary to popular belief, the game is actually very complex when trying to out-guess an intelligent opponent... authors of the top entries, including some well-known AI researchers, have commented that writing a strong [RPS] program was much more challenging than they initially expected" [25].

In RPS, if one player pursues a GTO strategy the payoff for both players in the game is fixed: if Player A plays ⅓ rock, ⅓ scissors, ⅓ paper, any strategy chosen by Player B becomes an equilibrium strategy. In most games, including poker, this is not true: a player who deviates from an equilibrium strategy in an attempt to increase their expected payoff runs the risk of counter-exploitation. This is the fundamental challenge of opponent modelling: how does an agent successfully exploit an opponent's mistakes while minimizing their own exploitability?

Poker AI research has seen considerable success in developing unexploitable strategies, with the solution of Limit Hold'em in 2015 and recent advances in No Limit Hold'em, but opponent modelling in poker has advanced much more slowly [3]. Research on the topic has focused on toy games or heads-up Limit Hold'em, which holds little interest for competitive players wondering how to apply this to their own game.

This section gives a brief overview of this research before analyzing the topic from the perspective of bot detection for the first time and applying this material to problems faced by human poker players.

4.1 Explicit agent modelling

Most research into opponent modelling consists of what Bard, in his comprehensive overview of opponent modelling techniques [26], describes as "explicit agent modelling": constructing and responding to a generative model of the opponent's behaviour from observing their play. A brief tour of this is worthwhile to set up later material.

One of the earliest treatments of the issue was given by Davidson et al. [27] and forms the basis for later work by colleagues at the Computer Poker Research Group. Their example uses a table to store the observed frequency of each action by a player; these frequencies determine the median hand strength for each action, which is used to calculate the *a posteriori* probability that the player holds each hand. A very simple model is used in the example but this can be extended to the desired level of complexity. This involves deciding how specific the action categories should be: a bet from early position suggests greater strength than a bet from late position so we should not lump together all betting actions, but using too many categories results in each having too few observations to be useful. This is analogous to the action abstraction decisions made when developing our own strategy and the choice of statistics made in the analysis in Section 6. The merit of this approach is that it scales well. A player with a deep understanding of poker can create a complicated system of categories and a novice player can follow a very simple system that requires no expert knowledge. The simple model given performed as well as a simple expert-defined model in a heads-up comparison. Writing in 2000, Davidson et al. argued that the calibration needed to refine the model is "laborious, and not particularly interesting from a scientific point of view". Since then, HUDs and the likes of PokerTracker have made this work much easier for players; it certainly is interesting from their point of view.

Hoehn et al. [28] apply Bayesian parameter estimation to Kuhn poker, starting with a pre-made profile of the opponent and computing the maximum *a posteriori* estimate for the parameters given observations of their play, to investigate the exploration-exploitation tradeoff. They find that shifting to exploitation is preferred after a surprisingly small number of hands. It is not guaranteed that this scales linearly to larger games; even if it does, the corresponding threshold for a game as large as No Limit Hold'em may not occur quickly enough to be useful, especially for bot detection. The authors observe that strategies may be equally exploitable yet differ in how easy they are to explore and suggest further study on this topic. In the decade since, this has been covered in the extensive literature on opponent modelling. However, the related point made in Section 3 - that strategies can be equally exploitable with some being easier for humans to understand - has not. More work is needed to see if there is a correlation between faster exploration for bots and easier comprehension for humans. Advances in this area will bridge the gap between academic research into poker AI and poker theory as applied in real games; actionable advice to human players for real-time exploitation would be very popular.

An early paper from Johanson [29] shows an attempt to retain the exploitative potential of frequentist approaches without the risk of a large loss if the opponent model is wrong. The Restricted Nash Response (RNR) method imagines a game in which the opponent must play an assigned unbalanced strategy with probability pand play the game normally with probability $1 - p$. The agent takes a stance on the tradeoff between exploitation and exploitability by varying p: values close to 0 give -equilibrium strategies (assuming that the opponent defaults to an equilibrium strategy in the 'normal' game) while values close to 1 give maximally exploitative strategies. Testing showed that, against a diverse set of computer opponents, this approach could successfully exploit the targeted opponents without surrendering much exploitability. This approach is more robust but is still reliant on a large number of opponent observations. As Johanson shows in a follow-up paper [30], with insufficient data increasing p does not guarantee superior performance over the equilibrium strategy but still leaves the agent more exploitable. Additionally, if the strategy used for the opponent model never reaches a certain part of the game tree, the default strategy is used instead; this can look very different from the RNR strategy, leading to poor decisions.

He proposes a natural alternative in the data-biased response ("DBR"). When choosing p in the RNR method we are implicitly making a statement about our degree of confidence in the model. Instead of having one fixed value for p, we can assign and vary this value for each information set to reflect our confidence in each part of the model. Generalizing RNR: when the opponent reaches information set i, they must follow the model with probability Pconf (i) and follow their normal strategy with probability 1 - Pconf (i), where Pconf represents our confidence in the model at that information set based on the number and type of observations. Again, this work assumed that the default opponent is the nemesis. Tests showed that this outperformed RNR, requiring fewer observations and less strict requirements about the input data, but is still not feasible to compute in a realistic time frame for large variants.

In the same vein, Ganzfried introduces a deviation-based best response ("DBBR") algorithm that accelerates opponent modelling by comparing the opponent's action frequencies to an equilibrium strategy. Empirically, this put up good results, but there are unresolved theoretical questions. The difference between the opponent model and the equilibrium is quantified in terms of hand strength: in the example given, if the equilibrium raises 50% of the time when the opponent only raises 30% of the time, the 20% of hands at the bottom of the equilibrium's range can be removed. As discussed later, this approach to range construction does not reflect how players do or should play poker. However, a similar algorithm can be developed that takes this into account, albeit imperfectly.

These explicit agent modelling methods take different routes around the same problem: the unreliability of models using small samples and the computational cost of generating full strategies. Bard advocates implicit agent modelling, in which an offline portfolio of expert strategies is used to estimate the utilities of actions, on the grounds that it avoids these theoretical issues and gives better empirical results. For a full treatment of this and recent developments, see Bard [26]; for a more detailed history of explicit modelling and other methods see Rubin and Watson [31].

The poker community has already embraced a type of frequentist analysis that may be able to sidestep these problems, as seen later in this section.

4.2 Neural networks

Davidson gives a prescient argument for seeking an autonomous system rather than one prone to human error with a study conducted using an artificial neural network [32]. Hand histories were used as input for the network with a variety of parameters, with the output predicting an opponent's next action much more accurately than a frequentist approach. Davidson expressed concern about whether an approach using neural networks would be feasible for real time use; seventeen years later, the success of the DeepStack No Limit Hold'em program that uses neural networks in early testing against humans suggests that it is. He investigates this further in a later work [33], using a simple neural network with decent results. He argues that, despite their possible effectiveness, they are theoretically ill-suited to the task because they are trained on actual events rather than the probability distribution, possibly skewing the output based on variance in the training data. However, they are useful as a way of identifying which features characterize common styles of play; this knowledge can be used with other opponent modelling methods for better results. For a summary of other early work using neural networks and other machine learning techniques, refer again to Rubin and Watson [31]

In general, neural networks are a likely source of developments in poker AI. PokerSnowie, a popular training tool and descendant of the Snowie program that became ubiquitous in online backgammon, was built using a neural network [14]. Importantly for this work, they are also a promising and largely unexplored way to develop poker AI with the goal of emulating human play [34][35]. A developer could use datamined hand histories as training data to build a replica of the 'average' player in the target game or use their own hand histories as training data to give the closest realistic approximation to their own play [36]; a comparative study of a player P and the corresponding bot P-NN is an ideal setting for testing methods for detecting bots. Observable differences between P and P-NN make reliable indicators of bot use and should also apply to other types of bot. Seemingly unique features of human play - deterioration due to fatigue or frustration, erratic adjustment to opponents' perceived weaknesses - are present in the training data, so they will appear in the bot's play. However, we expect P to display these features inconsistently and eventually whereas they will show up in P-NN's play immediately and uniformly. Replicating this inconsistency as a bot developer is a difficult task, so examining how an account's play changes over time is likely to remain a good test of authenticity.

An interesting opponent modelling question for future machine-machine contests (such as the Annual Computer Poker Competition [8]) is how methods for building bots leave different signatures in their play. For instance, if we think bots built using neural networks tend to have a specific flaw, we want to know if an opponent is one of these so that we can exploit this flaw.

4.3 Teams of agents

A twist on the frequentist approach is developed by Maitrepierre et al. [37] using a Bayesian method to update the probability of each possible holding for the opponent given their prior actions and their actions in the hand. The opponent is compared to a set of precomputed models representing a range of common playstyles (such as tight-aggressive or loose-passive). Based on this, the agent works out which model is most effective against the closest model to the opponent and adds it to their own repertoire. The opponent's play is monitored over a short interval and compared to the opponent model; if they seem to be adjusting their play, this calculation is done again so that the agent can adopt a new profile more quickly. The authors note that their method is computationally inefficient even for Limit Hold'em, limiting its relevance to competitive poker. They explain its average performance in the Annual Computer Poker Competition by noting that it performs especially poorly against strategies that are close to optimal; in games where humans play a wide range of highly suboptimal strategies, this is less of a concern.

A similar idea is given by Johanson et al. [29] who show that selecting from a team of agents, with a given strategy as one member, can yield better results than using that strategy exclusively. This was corroborated in the Man-Machine Match held between the CPRG's Limit Hold'em bot, Polaris, and two human experts. These experts remarked that, despite knowing they were facing a bot, they could not tell if the bot was adaptive and the threat of adaptation forced them to play differently.

One merit to this idea is that it disrupts opponents' attempts to model the agent. To be effective a strategy must form a cohesive whole: the actions taken in one situation determine actions taken in other situations, and so on. An opponent who gains some information about part of the strategy can fill in some of the blanks. If a team of agents is used, parts of the agent's combined strategy can shift suddenly and repeatedly over the course of the session. Consider a basic team consisting of an aggressive agent, a passive agent, and an passive agent who is more aggressive in certain situations to exploit common weaknesses. Depending on the change point detection technique used the choice of agent will change at various points. Observations gathered before each point will give a misleading picture of the agent's strategy afterwards: a seemingly passive agent has actually become an aggressive agent. This difficulty in modelling adjustment over time is familiar from the earlier discussion of neural networks and trying to have bots emulate human behaviour.

Even if the opponent knows that a team is used and that it consists of these agents it is unclear how to use this information: if they observe the agent taking more aggressive actions and believe this is a recent trend, it is unclear whether they are observing likely actions from an aggressive agent, exploitative actions from the exploitative passive agent, or simply actions from the narrow range of aggressive actions taken by the passive agent. In general, if these strategies are sufficiently different, any attempt to model the combined agent as a single entity will give a model that is very different from how the agent is actually playing at any given time. This can be seen in Section 6, where the statistics for the composite bot player were noticeably different from the statistics of any one bot. To make an informed guess the opponent needs an idea of the agent's change point detection technique and method for choosing an agent from the team, which are both hard to determine just from observing of the combined agent's play. This requires second-order opponent modelling in which the opponent models its own strategy, estimates how the strategy is being modelled by the opponent, and then makes a best guess as to how the opponent is adjusting. Each step is increasingly difficult for human agents. As discussed in Section 3, an agent can execute a strategy well without being able to describe it formally and can describe it formally without being able to execute it well. The difficulties humans face in modelling opponents suggest that modelling how they are being modelled will be even tougher. If a human suspects they are facing a bot, the survey answers in Section 3 suggest there is clear disagreement about and no clear understanding of whether and how bots can and do exploit opponents. If a human doesn't know whether an opponent is a human or a bot, this adds another layer of difficulty. Finally, the difficulties inherent in opponent modelling and specific to humans trying to model opponents are well-documented, both elsewhere and in this section.

We can extend the idea of an equilibrium strategy to teams of agents: a strategy by itself may be very unbalanced but, as part of a team where each strategy is chosen with the appropriate frequency, the combined strategy can be balanced. If both players field teams of agents exploitation can consist of adjusting the frequency or method for choosing the current player rather than modifying the strategy in real time. For large-scale bot construction, this approach imposes a heavy upfront cost in computational effort - multiple strategies have to be computed - but potentially reduces the challenges of in-game adaptation as no real-time adjustments are needed.

It might help to frame other poker strategy questions in this way. When a human player adjusts their strategy to exploit an opponent, this can be thought of as selecting from a team of many agents with 'close' strategies differing only in a few ways. Fundamentally, what we refer to as a "strategy" - a distribution over the possible actions at each information set in the game - can be seen as a choice, at each information set, from a team of agents representing pure strategies for each action. In turn, the process guiding this choice can itself be seen as a member of a team - a process that tends to select aggressive actions is treated as an aggressive agent, and so on - and this continues up each level of the game tree.

This team-based approach offers a useful framework for thinking about the bot detection problem. When building a team of agents we want them to 'cover' as wide a strategy space as possible between them so that, no matter what the opponent's strategy, we can choose a strategy that minimizes exploitability or maximizes exploitation against it. Limits on computational resources may cap the number of agents in the team; the team manager has to maximize the area covered in the strategy space under this and other constraints. Similarly, as discussed in Section 6, a bot ring operator must ensure that the distances in this space between any two members and between individual members and the average player are within the usual margins of error. The abrupt changes in strategy described above will mark a player as a bot so members have less freedom to vary their own strategies (the bot ring described in Section 5.5 is an amusing example: the bots received a software update that radically altered their play but this was introduced all at once and applied to all bots at the same time, making their identity as bots glaringly obvious). However, the ring as a whole can be considered a team at the level of game selection with the role of manager played by the bot ring operator, who must also apply these game-theoretic considerations to other problems. For instance, we want our bots to play for as long as possible to maximize our earnings but alarm is raised if a player plays for too long in one session, two players are at the same table too often, or players join or leave at the same time; given these constraints, how should we distribute the bots' participation in games?

4.4 Detecting suspicious activity

Mazrooei and coauthors from the CPRG use poker as a case study to test their approach for detecting collusion in games. A collusion table tracks the effect of each player's actions on the utility of each other player. CFR is used with two abstractions of the game to generate base agent functions that define the value. The utility functions of suspected colluders are modified to include each other and two measures - the agents' combined impact on their total utility and the marginal impact of one agent on another's utility compared to regular agents - determine if a player is colluding. The results show that colluders are clear outliers, which matches the findings of similar real-world investigations [38]. It is not surprising to learn that this also requires a prohibitively large sample size, taking 90,000-100,000 hands to classify the strong colluders. The authors argue that this is not large compared to the total number of hands a high-volume regular might play in a year but collusion brings a strong incentive to detect it quickly: players who are victims of apparent collusion before it is formally detected will lose their money and their confidence in the site.

Opponent modelling for the sake of exploitation usually entails trying to model the opponent's full strategy, presenting serious complications. However, we can use the same techniques for more narrow tasks. Suppose a player starts to worry that one or more of their opponents is a bot. As discussed, we would expect a bot's play to be discernibly different from a human's. Building a full opponent model may be impossible but the agent is not trying to exploit the bot, just detect it. If this only requires a partial opponent model that captures specific behaviour, the task becomes easier and we may have access to previously unviable modelling techniques.

If we know which behaviours are red flags, it may be simpler and more effective to just look for instances of these. If the population contains known bots, we can use a supervised learning technique to give a classifier that can automatically label suspects [39]. In practice, most of the time we have a vague suspicion and no immediate proof, so we have to look elsewhere. One idea is to adapt the team of agents approach. We construct a set of bots representing the range of common strategies in the game and adjust them to exhibit the target behaviour; if we are worried about collusion, we teach the bot to collude with minimal disruption to its overall strategy. We then look for others in the game who follow the same basic strategy as a baseline group. If an account following the strategy is 'closer' to the collusive bot than to the baseline player, it deserves a closer look. In their fraud investigation Stephens-Davidowitz and Bakker question the wisdom of a similar idea: "The problem is that if an AI cheats relatively subtly, it would be extremely unlikely to find a real cheater with the same pattern of obfuscation. If it does so blatantly, then its play will also have a lot of other differences from the play of a non-cheating AI that would not necessarily indicate cheating - it would, for example, bluff and slowplay much more often" [38]. This is certainly true to an extent but not a reason to discard the idea. Part of 'subtle' cheating is that it is hard to detect: if our own bot is able to blend in while cheating, this can illuminate how other agents might do the same even if the cheaters in this specific case are different. This AI comparison approach does not have to be the sole determinant of a suspect's guilt but can be part of a series of tests that each have their own strengths. The idea of using bots to detect other bots has a pedigree in backgammon and has been suggested by AI researchers for poker but not deeply explored [14].

4.5 Cross-set tabulation

Previously, we asked how often a player takes the possible actions at each information set: for example, how often they fold, call, or 3-bet (reraise) when facing a preflop raise of size (x) with stack sizes (y, z) on the button while holding Ac2c. We then repeat this for each information set: what if the player holds Ac2h or AcKc instead, or the raise is larger, or the stacks are deeper? For large variants, abstraction methods are used to group information sets together to reduce the size of the game tree: preflop raises within a certain range are treated as the same size, or hands that are close in potential are played the same way [40]. These answers are then aggregated to give a profile of the overall strategy.

As Southey et al. argue in their discussion of priors for Bayesian opponent modelling, "The size of the game virtually guarantees that one will never see the same information set twice. Any useful inference must be across information sets and the prior must encode how the opponent's decisions at information sets are likely to be correlated" [41]. In this new approach, opponents are profiled from statistics that represent how often the same action is taken across a subset of information sets at the same level of the game tree: in total, how often does a player 3-bet preflop? How often do they call a continuation bet from middle position versus late position? These statistics form the basis for further analysis. The conceptual difference is illustrated in Figure 2. On the right, representing the traditional abstraction method, the vertical route through the game tree determines which unique instance of Raise (R) or Call (C) is reached; on the left, representing this new approach, we examine one horizontal level of the game tree to find all R actions or all C actions.

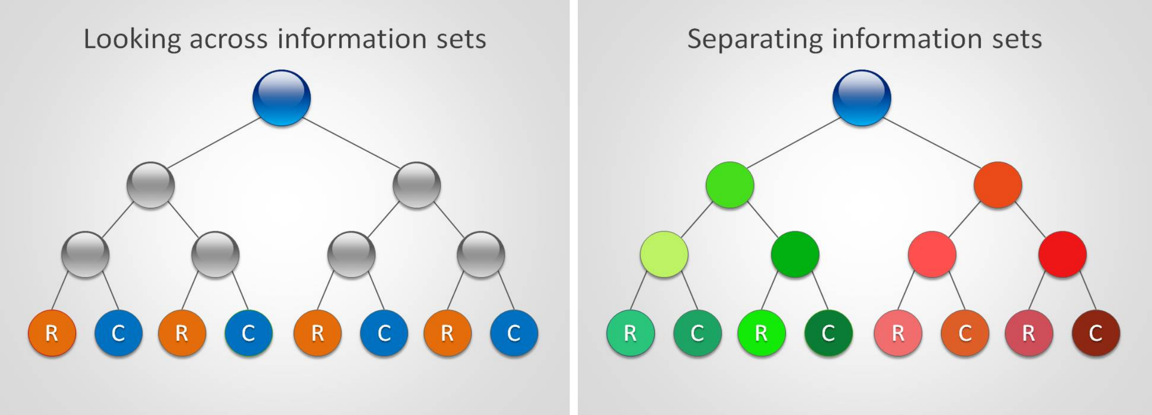


Figure 2: Cross-set vs individual information sets

This approach is agnostic to other features of the information set and does not need to know the route taken through the game tree to arrive at it (except insofar as this is part of the feature); by contrast, the former approach does not care about what the information set 'means', just the route taken and possible future routes. To illustrate this, consider this abridged series of actions:

Preflop:

- Agent raises to 3 BBs
- Opponent calls

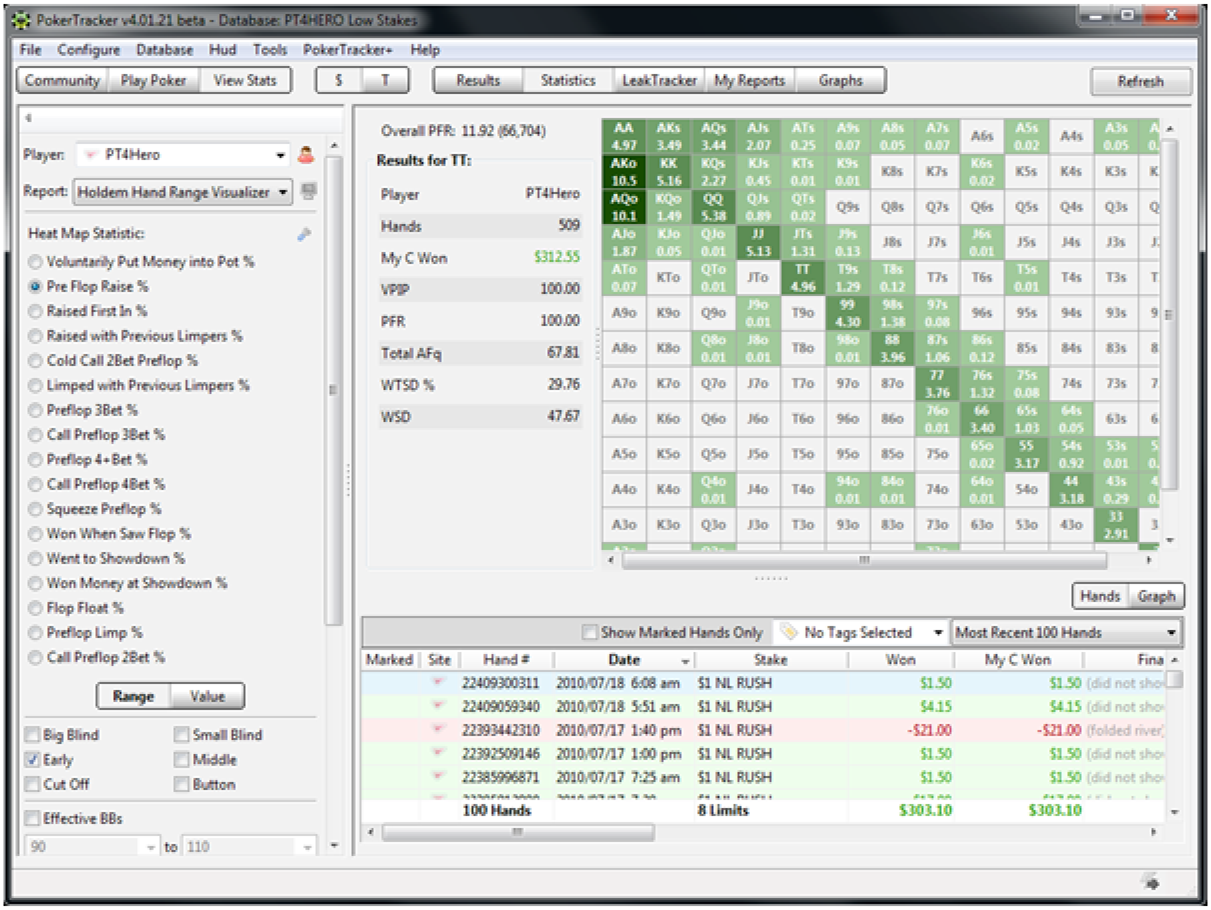
Flop:

- Agent makes a continuation bet ("c-bet")

The 'vertical' approach uses the actions to determine its next path: the agent chose to raise, and raised to this specific amount, so follow this path; the opponent called instead of reraising, so follow that path. If all labels were removed from the game tree, nothing would change; the distinguishing feature of a state is just its position in the larger path. For the 'horizontal' approach, suppose we want to know how often the agent bets if the opponent reraised preflop. We go back one step in the game tree to see if this instance fits the criterion (the opponent reraising), but beyond this we do not care how big the initial raise or reraise were. Since this approach repeats this process for many conditions, which care about different levels of the game tree, it is possible for each node in a path to be evaluated without the path itself being evaluated *qua* path in its entirety.

This has several advantages. We do not need all parts of a strategy to show up frequently in the sample for these statistics to accurately reflect the true shape of the strategy, though they will still be skewed if the sample is not representative. When trying to model the opponent in real time this method ensures that all observations can contribute meaningfully to the model (unless there is a specific cause for concern), requiring a smaller sample. A good approximation of each statistic can be reached with little computational effort.

Following the analysis in Section 3, this approach is much simpler for humans to understand: rather than needing a holistic view of a distribution of actions over an impossibly large game tree, they just need to know the significance of a few statistics. This makes it easier to recognize irregular behaviour: a VP$IP or PFR value that is much higher than expected will register as odd more quickly than a pattern of behaviour only visible across many information sets. As a result, poker analysis tools often present information in this form as shown in Figure 3.



Player: PT4Hero

Report: Holdem Hand Range Visualizer

Heat Map Statistic:

○ Voluntarily Put Money into Pot %
● Pre Flop Raise %
○ Raised First In %
○ Raised with Previous Limpers %
○ Cold Call 2Bet Preflop %
○ Limped with Previous Limpers %
○ Preflop 3Bet %
○ Call Preflop 3Bet %
○ Preflop 4+Bet %
○ Call Preflop 4Bet %
○ Squeeze Preflop %
○ Won When Saw Flop %
○ Went to Showdown %
○ Won Money at Showdown %
○ Flop Float %
○ Preflop Limp %
○ Call Preflop 2Bet %

□ Big Blind	□ Small Blind
☑ Early	□ Middle
□ Cut Off	□ Button

□ Effective BBs

90 to 110

Overall PFR: 11.92 (66,704)

Results for TT:

Player	PT4Hero
Hands	509
My C Won	$312.55
VPIP	100.00
PFR	100.00
Total AFq	67.81
WTSD %	29.76
WSD	47.67

AA 4.97	AKs 3.49	AQs 3.44	AJs 2.07	ATs 0.25	A9s 0.07	A8s 0.05	A7s 0.07	A6s	A5s 0.02	A4s	A3s 0.05	A 0.
AKo 10.5	KK 5.16	KQs 2.27	KJs 0.45	KTs 0.01	K9s 0.01	K8s	K7s	K6s 0.02	K5s	K4s	K3s	K
AQo 10.1	KQo 1.49	QQ 5.38	QJs 0.89	QTs 0.02	Q9s	Q8s	Q7s	Q6s	Q5s	Q4s	Q3s	Q
AJo 1.87	KJo 0.05	QJo 0.01	JJ 5.13	JTs 1.31	J9s 0.13	J8s	J7s	J6s 0.01	J5s	J4s	J3s	J
ATo 0.07	KTo	QTo 0.01	JTo	TT 4.96	T9s 1.29	T8s 0.12	T7s	T6s	T5s 0.01	T4s	T3s	T
A9o	K9o	Q9o	J9o 0.01	T9o	99 4.30	98s 1.38	97s 0.08	96s	95s	94s	93s	9
A8o	K8o	Q8o 0.01	J8o 0.01	T8o	98o 0.01	88 3.96	87s 1.06	86s 0.12	85s	84s	83s	8
A7o	K7o	Q7o	J7o	T7o	97o	87o	77 3.76	76s 1.32	75s 0.08	74s	73s	7
A6o	K6o	Q6o	J6o	T6o	96o	86o	76o 0.01	66 3.40	65s 1.03	64s 0.05	63s	6
A5o	K5o	Q5o	J5o	T5o	95o	85o	75o	65o 0.02	55 3.17	54s 0.92	53s 0.01	5 0.
A4o	K4o	Q4o 0.01	J4o	T4o	94o 0.01	84o 0.01	74o	64o 0.01	54o	44 3.18	43s 0.29	4 0.
A3o	K3o	Q3o	J3o	T3o	93o	83o	73o	63o	53o	43o	33 2.91	3

□ Show Marked Hands Only   No Tags Selected   Most Recent 100 Hands

Marked	Site	Hand #	Date	Stake	Won	My C Won	Fina
		22409300311	2010/07/18 6:08 am	$1 NL RUSH	$1.50	$1.50	(did not sho
		22409059340	2010/07/18 5:51 am	$1 NL RUSH	$4.15	$4.15	(did not sho
		22393442310	2010/07/17 1:40 pm	$1 NL RUSH	-$21.00	-$21.00	(folded river)
		22392509146	2010/07/17 1:00 pm	$1 NL RUSH	$1.50	$1.50	(did not sho
		22385996871	2010/07/17 7:25 am	$1 NL RUSH	$1.50	$1.50	(did not sho
		100 Hands		8 Limits	$303.10	$303.10	

Figure 3: Popular poker analysis software PokerTracker

The disadvantage of ignoring these particularities is that we lose the ability to characterize instances of an action in useful ways. For instance, we may know that the opponent 3-bets preflop 10% of the time but this does not tell us which hands make up that 10%. 3-betting is an aggressive action signifying strength but an opponent who 3-bets exclusively (or exclusively 3-bets) with their strongest hands will be predictable and exploitable. A smart player will distribute their strongest hands between their ranges for each available action and fill in the gaps with a suitable mix of other hands. Following the observation that there can be many strategies with the same level of exploitability and these can vary widely, we note that this variation consists largely of how the agent selects which hands go in which ranges. A player might have a highly polarized 3-betting range consisting of just strong hands and weak hands or a 'flatter' range where the remainder is made up of medium hands as well as weak hands. Suppose our sample only has an opponent 3-betting 10% of the time but they frequently do this with weak hands. Should we assume that the sample is accurate and these observations consistent with a polarized range, or is this part of a flatter 3-betting range from a player who actually 3-bets more often than the sample suggests? In contrast, a sample in which they mostly 3-bet with strong hands does not prompt these questions.

Similarly, we might think that certain actions indicate something about the shape of the strategy elsewhere. Poker theory has shown that an optimal strategy should be able to make a wide range of bet sizes, including large overbets [10]. In practice, few players do this and those that do tend to be either excessively aggressive players or skilled players who can recognize when this is appropriate (or both). Which of these explanations we are drawn to depends on whether how much aggression we observed from this player in other contexts. This suggests using this method in conjunction with qualitative analysis to make sense of these findings. We will now see examples of how this is successfully used.

5. Analysis of independent investigations

A challenge for poker players trying to investigate fraud of any kind is finding a useful precedent. Most cases are dealt with in internal investigations by poker sites with no public engagement. Sites are reluctant to acknowledge problems unless necessary; in a rare exception, 888poker published a revealing guide to identifying bots [42]. Discussions on public forums are quickly overwhelmed by misinformed or misleading comments; unsubstantiated accusations of bot use are common, casting doubt on legitimate concerns. In many cases, when the issue is resolved there is no follow-up so observers are unsure if the suspects are still active. No formal analysis of these investigations is given so they are treated as isolated events. Looking for patterns in the observations that trigger investigations, how they are conducted, and when they are successful is a worthwhile topic for future work; an initial attempt is made here.

Two comprehensive fraud investigations by the same author are used to show how statistical analysis can be applied to these problems. Two basic examples of bot detection efforts on the largest poker forum give the context for how this is often done in practice. A third, more detailed bot detection effort is used in conjunction with these to draw general conclusions about the merits of this approach and highlight specific areas to focus on.

Images are taken from the respective write-up by each author.

5.1 Cake Poker security breach (2010)

In this study of potential fraud resulting from security vulnerabilities on the Cake Poker network, Stephens-Davidowitz and Bakker were contracted to detect players who benefited in some way from accessing opponent's private cards [38]. A 'value' was computed for each hand as a function of the agent's and opponent's hand strengths and how aggressive the opponent's play is (since having this information strongly encourages aggressive play; they note that the choice of aggression multiplier was somewhat arbitrary, as we did for our own choice of multiplier in Section 6. Each pair of players that were in a hand together was examined for outliers. As a backup measure, players were singled out if they had taken river actions incongruous with the strength of their hand (such as calling a large bet with a weak hand) and the rate of these actions was compared with the average player.

Cake Poker's security issue had become notorious and there had been a series of widely publicized scandals involving private card access on other sites, so it was assumed that cheaters would try to avoid detection. The authors expected that, "if a cheater were sufficiently careful, he could have easily earned large amounts of money quickly while still appearing completely unremarkable to manual inspection, winrate analysis, and many other naive means of detection". In fact, players intentionally losing to their partners were easily caught even though they "went to fairly extreme lengths to hide the fact they were chip dumping, taking hundreds of hands to move relatively small amounts of money between accounts in a way that looks very much like normal play". They cite a player whose "play looks qualitatively well within the range of normal, and he made numerous plays that were not consistent with his goal of chip dumping… however, using our main method of detection, he was clearly an extremely blatant outlier… nobody who didn't cheat came anywhere close to being such a huge outlier".

This reinforces the arguments in Section 3 about the difficulties humans face when trying to model and implement strategies. The players colluding have to:

- know what the average loss rate is and understand how this is reached in normal play

- successfully implement this strategy while accomplishing the goal of chip dumping
- model the actions of the site security team trying to detect this behaviour

Players already struggle to understand and execute strategies for which they have a strong frame of reference; the problem of trying to lose money without alerting observers does not feature in most instructional material. Even if they have a practical and theoretical understanding of what ‘normal’ play looks like, they are unlikely to know which deviations from it will attract attention and have no way of knowing what methods the site operators are using to detect them. The player described made a valiant effort and was still caught very easily by a model that its creators did not expect to work.

5.2 Collusion investigation (2010)

A well-known professional poker player was accused of multiaccounting (playing under different aliases on the same site in violation of the Terms of Service). Meanwhile, a number of other accounts were suspected of colluding by intentionally playing passively against each other. The two cases were linked, prompting an investigation [43].

There were clear differences between each player’s actions against a general population and their actions against each other. In particular, the suspects would raise a lot pre-flop and fold often to 3-bets creating an incentive for other players to 3-bet them more frequently; instead, the suspects 3-bet each other much less often than other players, at a highly suboptimal rate. The example shown in orange on Figure 4 (taken from the report) represents a 3-betting range of only three combinations of hands - AA, KK, AK - which is highly suboptimal and lower than the 3-betting frequency of any regular player. An otherwise competent player making such a sudden and egregious error in specific situations is telling.

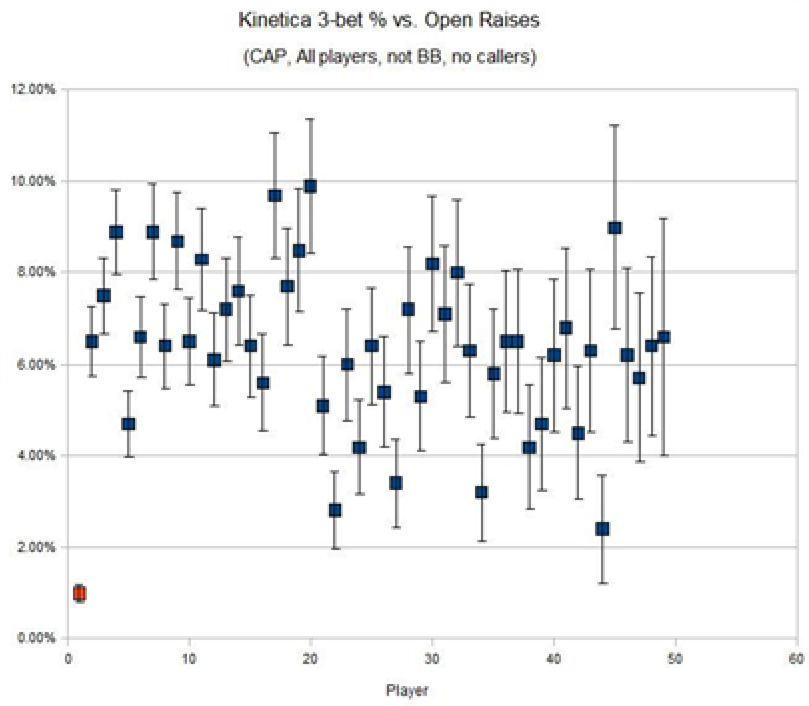


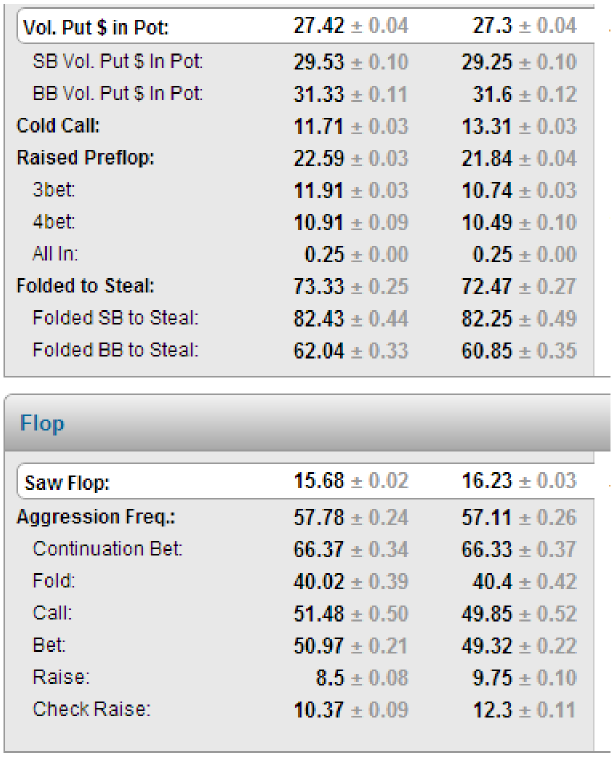
Figure 4: Colluder as an outlier against other players

5.3 PartyPoker (2014)

Two accounts were identified as bots based on shared and unusual behaviour patterns [44]:

- ‘Regular’ bet sizes: multiples of 5
- Abnormally high 3-bet%
- Near-identical winrate over a large sample
- Erratic pre-flop opening sizes

Figure 5 shows the striking similarity between the two accounts:



Vol. Put $ in Pot:	27.42 ± 0.04	27.3 ± 0.04
SB Vol. Put $ In Pot:	29.53 ± 0.10	29.25 ± 0.10
BB Vol. Put $ In Pot:	31.33 ± 0.11	31.6 ± 0.12
Cold Call:	11.71 ± 0.03	13.31 ± 0.03
Raised Preflop:	22.59 ± 0.03	21.84 ± 0.04
3bet:	11.91 ± 0.03	10.74 ± 0.03
4bet:	10.91 ± 0.09	10.49 ± 0.10
All In:	0.25 ± 0.00	0.25 ± 0.00
Folded to Steal:	73.33 ± 0.25	72.47 ± 0.27
Folded SB to Steal:	82.43 ± 0.44	82.25 ± 0.49
Folded BB to Steal:	62.04 ± 0.33	60.85 ± 0.35

Flop

Saw Flop:	15.68 ± 0.02	16.23 ± 0.03
Aggression Freq.:	57.78 ± 0.24	57.11 ± 0.26
Continuation Bet:	66.37 ± 0.34	66.33 ± 0.37
Fold:	40.02 ± 0.39	40.4 ± 0.42
Call:	51.48 ± 0.50	49.85 ± 0.52
Bet:	50.97 ± 0.21	49.32 ± 0.22
Raise:	8.5 ± 0.08	9.75 ± 0.10
Check Raise:	10.37 ± 0.09	12.3 ± 0.11

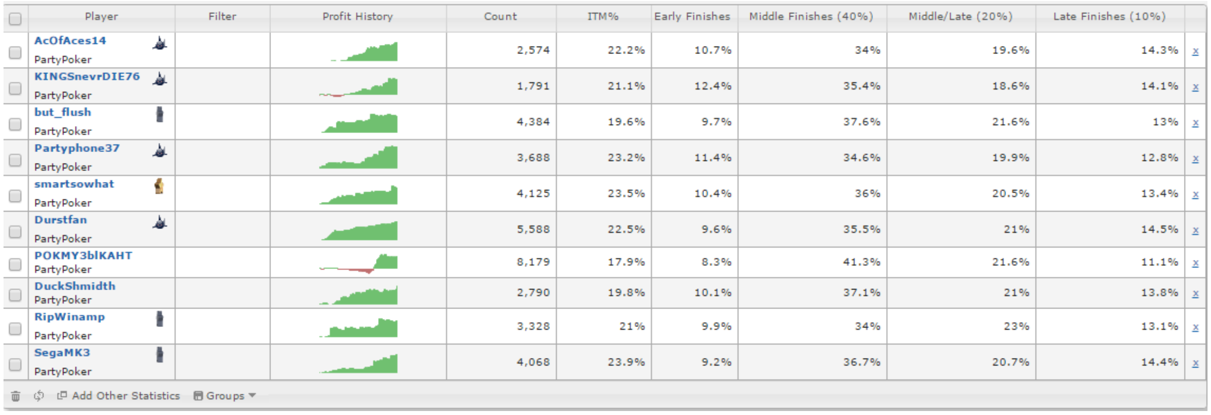
Figure 5: Statistics for two bots from the same ring

5.4 PokerStars/PartyPoker/Full Tilt Poker (2016)

A new group of regular players appeared in tournaments across three major poker sites, winning over a million dollars in profit [45]. Again, these were identified as bots based on similar play:

- Their distribution of tournament finishes (early/middle/late) followed the same pattern (see image)
- An almost identical and very high W$WSF statistic (around 50% compared to an average of around 45%)
- 'Contradictory' statistics: high aggression but passive against opposing aggression
- No awareness of tournament-specific considerations such as the changing structure or ICM calculations

Note the appearance of W$WSF as an important statistic. The tournament format is more difficult to engineer a bot for as the structure of the game - blinds and possible antes - periodically changes. A competent tournament player has to adjust to this so a bot hoping to pass as a human must try to as well.



□	Player	Filter	Profit History	Count	ITM%	Early Finishes	Middle Finishes (40%)	Middle/Late (20%)	Late Finishes (10%)	
□	AcOfAces14 PartyPoker			2,574	22.2%	10.7%	34%	19.6%	14.3%	x
□	KINGSnevrDIE76 PartyPoker			1,791	21.1%	12.4%	35.4%	18.6%	14.1%	x
□	but_flush PartyPoker			4,384	19.6%	9.7%	37.6%	21.6%	13%	x
□	Partyphone37 PartyPoker			3,688	23.2%	11.4%	34.6%	19.9%	12.8%	x
□	smartsowhat PartyPoker			4,125	23.5%	10.4%	36%	20.5%	13.4%	x
□	Durstfan PartyPoker			5,588	22.5%	9.6%	35.5%	21%	14.5%	x
□	POKMY3blKAHT PartyPoker			8,179	17.9%	8.3%	41.3%	21.6%	11.1%	x
□	DuckShmidth PartyPoker			2,790	19.8%	10.1%	37.1%	21%	13.8%	x
□	RipWinamp PartyPoker			3,328	21%	9.9%	34%	23%	13.1%	x
□	SegaMK3 PartyPoker			4,068	23.9%	9.2%	36.7%	20.7%	14.4%	x

Figure 6: Unusually similar distribution of tournament finishes between bots

5.5 PokerStars/PartyPoker (2015)

This is the most detailed and scientifically rigorous investigation of suspected bots in a public forum; it motivated the test in Section 6 and is a useful blueprint for future investigations [46]. A central figure in this discussion generously answered questions about their approach, informing the analysis below [47].

Players in high-stakes Pot Limit Omaha games on PokerStars reported suspicious behaviour from some opponents. A selection of play statistics, both generic (such as the familiar VPIP and PFR) and specific (such as check-raising frequency on the river), were computed for these accounts; these were chosen to cover a wide range of possible areas of difference between suspects and other players. As regular players in these games, the authors also used their own statistics as a basis for comparison. Accounts were initially grouped based on which statistics saw two accounts differ by less than a 2% margin and then a 5% margin; this was enough to identify a first round of ten accounts, which were purged from the site. The second round saw the use of a basic Euclidean distance measure between these statistics to quantify the similarity between accounts (see Figure 7). Analysis of their participation showed that this ring had played almost eighteen million hands and caused over four million dollars in damage to players.

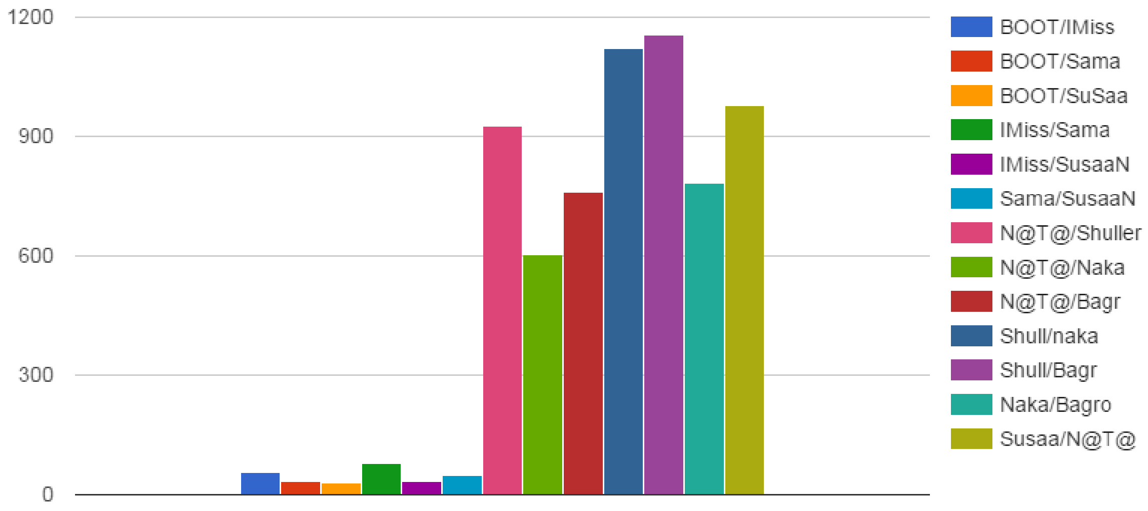


Figure 7: The first six columns compare bot pairs; the next six, human pairs; the last column, a bot and a human

The detail of the investigation gives us many areas of similarity between bots and difference between bots and humans; some are specific to this case, but most can be generalized:

- 'Contradictory' play in rare situations: these bots 4-bet much more often than humans but folded to another reraise much often than they should.

Greve suggests that this is emblematic of a bot hard-coded to follow a certain strategy that has not been given a more granular strategy for rare situations. When developing a manually coded bot, some parts of the game tree are visited rarely enough that it is not worth the effort required to make a marginal improvement to the strategy at that point. These situations are a valuable opportunity to observe unusual play from opponents. Because of their rarity statistics for these situations are based on fewer observations and are less reliable, but these are also more likely to be situations like this one where fewer observations are needed to know that something is amiss.

This inconsistency is worth noting in more common situations too. Suspects had around a 10% squeeze rate from the big blind, which is suspicious by itself as even the most aggressive players are capped at around 7%, but they did not display other play patterns expected from a player this aggressive.

- Very similar and consistent ranges when this is unlikely *a priori*: the bots followed the same pattern of check-raising a lot on the flop and turn but rarely on the river, and the flop and turn ranges were almost identical for each bot; in addition, these ranges were very similar between bots (check-raising frequency of 14-15% on the flop or turn and 2-3% on the river)

This is generally suspicious but check-raising ranges are especially volatile. Small adjustments to a strategy or variations in how a player follows that strategy can cause big swings in check-raising frequency. It is common for players with very similar play styles to have this differ by multiple percentage points; this level of consistency across multiple accounts is startling.

Likewise, a player's betting and check-calling frequencies are useful indicators of the shape of that player's overall strategy and a change to the strategy will manifest in these statistics. The suspects displayed a remarkable consistency here as well.

- Suspects raised when folded to pre-flop at a nearly identical rate and this similarity was observed across all positions

How often a player 'opens' is suggestive of which hands are in their range and therefore how they will tend to play on future streets. We expect variation between players in how this differs based on their starting position: two players who open about as often from early position might disagree on how often to open from late position. These accounts maintained this similarity for each pre-flop position and the margins of difference for each position was very small; we would not this from most pairs of human players (and especially not from a group of five or more).

In addition, there were the usual examples of actions taken at a rate that, empirically, is rare for humans. Once again, W$WSF clearly distinguished the two sets of players: 49-52% for the bots and around 45% for most humans.

This analysis highlights how useful expert knowledge can be in determining how much weight should be assigned to different play patterns. It also suggests that identifying common discrepancies and testing for these is a useful shortcut when it is infeasible to compare players' full strategies.

6. Measuring distance between strategies

Poker site operators use a range of technical measures such as CAPTCHA and mouse tracking to detect suspicious activity. These measures vary widely in quality between sites and those perceived as having weak security are quickly identified; creative bot developers can avoid detection at this stage [48]. For these efforts to be effective, sites must supplement these measures with analysis of in-game behaviour. The UK Gambling Commission explicitly endorses "identifying players that an operator deems higher risk and periodically checking their gameplay statistics for unusual behaviour, for example where there are suspicious similarities between groups of individuals with high win rates, or where gameplay indicators stand out as being irregular" [49].

However, there is no formal analysis in the public domain of how this should be done. As shown in Section 5, independent investigations of suspicious activity often start by noting obvious cases of the target behaviour and trawling through large data sets to find similar examples. This can be effective on a case-by-case basis but a systematic approach to fraud detection requires a theoretical framework for modelling these differences. This section marks the first attempt to sketch such a framework.

6.1 Requirements

Most suspicious activity manifests in unusual play patterns. When two players intentionally collude each behaves differently if their partner is involved in the hand [50]. If multiple accounts are used by the same player their play will be abnormally similar. A player with access to opponents' private information will play differently even when trying to 'ignore' this information [38]. As argued in Section 3, bots are likely to have discernible differences from human players no matter how similar their strategies are - and these strategies are often quite different.

In each case we observe differences between a 'normal' style of play and the behaviour of the suspicious account. Each account, $P_1, ..., P_N$, follows a strategy $S_1, ..., S_N$. Each of these strategies must be different enough from one another to not seem unrealistically similar while remaining within an acceptable margin of a member of a family of baseline strategies $S_A, ..., S_K$ .. Assessing whether two strategies are too similar or different requires a distance measure for quantifying this difference. Probability theory offers a wide range of measures [51] and it is not obvious which is most useful for this purpose. The existing literature makes little mention of this problem despite its theoretical importance [52]. We will see that specific tasks have required a choice of distance measure and experimental comparisons have shown the merits of some measures, but a general treatment is needed.

This metric ought to satisfy important constraints. For detection avoidance, an agent wants to ensure the distance between individual elements of their strategy and the corresponding elements of the baseline strategy remains within a given margin (which will change between elements) in addition to the total distance being falling within a certain range. A strategy that closely resembles another strategy but differs greatly in one area still attracts attention.
The context in which the measure is needed may lead us to assign extra weight to certain elements and the measure should allow this. As above, some actions (e.g. large overbets) are part of an optimal strategy but are rarely taken by most humans. A small difference in how often a player takes these actions may be more significant than a larger difference for an action seen from most players. If we are looking for bots in a small population with known tendencies, there may be specific deviations made unusually often by the average player in the population (with a view to exploiting those tendencies, for instance) that will not be shared by a bot.

There are also qualitative reasons to care about how this difference is distributed between elements of a strategy. Following the analysis in Section 3, many human players have trouble balancing the frequency of their actions. Intuitively, players who suffer more severely from this problem are likely to be less skilled in other aspects of their play. If an account performs better in situations that humans handle poorly but does not perform significantly better overall, this should be flagged; however, since the overall difference is low, a measure that only cares about total distance or distances for individual elements will fail to recognize this.

For instance, the abstraction process used in research to create bots for large poker variants involves action abstraction (grouping similar actions and restricting the agent to choosing between these groups)[1]. This is necessary for bet sizing in Pot Limit and especially No Limit games where players choose from a near-continuous range of bet sizes over an interval. Most bots are assigned a small number of bot sizes and potential bets that fall into a certain range are mapped to the nearest bet size. In contrast, the typical human range of bet sizes is distributed [1] more erratically with clusters around sizes that are common or easy to calculate (such as ½ pot, pot) and rough approximations of other sizes. Modelling this tendency so that a bot can emulate it is a hard task. One approach is to make a quasi-random selection from bet sizes in a

[1] The inherent dangers of this are shown by the ‘off-tree’ problem encountered by Carnegie Mellon University’s agent Claudico in its high-profile match against top human players. Mapping a wide range of opponent’s bet sizes to a fixed bet size results in some bets being misinterpreted and the bot taking actions appropriate for a different bet or pot size [53]. Such deviations from an account’s normal play are potentially useful markers of bot activity. Early impressions from the second Man vs. Machine NLHE Competition, taking place as this work is being submitted, suggest that CMU’s new agent Libratus has found a way to address this problem.

range of the abstracted bet size - if a bet is mapped to n% pot, select from a weighted distribution in the range (n-x)% to (n+y)% for small x and y - but this distribution will still be ‘smooth’ in a way human play patterns are not. In practice, poker site interfaces contain buttons for pre-set bet sizes as a convenience for players. If there is a ‘½ pot’ button, a casual player wanting to bet roughly that amount is likely to use the button. A professional player willing to use and calculate a more precise bet size is likely to be a high-volume player for whom the marginal gain from varying bet sizes is not worth the distraction from decisions demanding attention on other tables. In both cases, bet sizes in the vicinity of ½ pot will collapse into ½ pot. A more even distribution of sizes like 48%, 51%, etc. is highly abnormal, but a distance measure based on a rigid numerical ordering of elements (in which 44% is closer to 45% than to 50%) fails to capture this. This is illustrated by examples of bots caught because their bet sizes rounded differently from the pre-set bet sizes calculated on-site: instead of betting (n) the bot bet (n +/- epsilon) with the same frequency, a clear sign of mischief even though the absolute distance is small.

For the sake of bot detection, this measure should still function when we have an incomplete picture of an opponent’s strategy because of a limited data set. As parts of the game tree are visited much more frequently than others, our confidence in our model will vary and the model should acknowledge this. This relates to the previous criteria: situations that come up rarely are harder for humans to play optimally [14] and therefore likely (and potentially more significant) sources of imbalance in a strategy. For large poker variants, even with a coarse abstraction and a large sample size, only a small fraction of the nodes in the game tree will ever be visited (and, of these, most will not be visited frequently enough to make useful observations). A measure that requires this full model to be useful is impractical for bot detection.

6.2 Choosing a measure

The exploitability of a strategy is an objective measure of its difference from an optimal strategy; extending this, we could use the difference in the exploitability of two strategies to measure their difference from each other. However, the exploitability of a strategy is relative to the value of the game; the process for computing this value is linear in the size of the game tree and therefore difficult for large poker variants. Methods have been found to accelerate this process for Limit Hold’em [54] but this remains unrealistic for even the smallest form of No Limit Hold’em [55]. This value is only well-defined for two-player games and is unsuitable for the multiplayer games in which most online play occurs. In addition, the exploitability value tells us little about the overall shape of the strategy. Imbalances in one part of the strategy may be compensated for elsewhere: compare a strategy that is generally balanced with a small number of large imbalances against a strategy with a large number of small imbalances. If a bot and human player have strategies with about the same exploitability, the bot's strategy may contain features that identify it as a likely bot.

The choice of distance metrics for other tasks in poker research may be informative. The most common application is quantifying distances between groups of hands for information abstraction. An early approach to this used L_2-distance without explaining this choice [56]. Testing showed that this is outperformed by the Wasserstein distance (also known as the 'earth mover's distance' or 'EMD'), representing the amount of 'work' needed to transform one distribution into another [40]. EMD is a sensible choice for information abstraction as the ranking of hands suggests a clear ordering (e.g. AA > AK > JJ). Recent work by Ganzfried and Yusuf "generalizes EMD to multiple distributions" [12] and Ganzfried suggests EMD as a general measure for evaluating distances between strategies [52].

The use of EMD has several pitfalls. For small one-dimensional distributions, EMD is easily calculated in linear-time. However, for large, multi-dimensional distributions, computing EMD is much more difficult (and considerably harder than computing (Lp) distances for the same data); the best known algorithm for computing EMD is too slow for its intended use in poker abstraction. Ganzfried introduces a method for approximating EMD that is fast enough for abstraction, where an EMD-based approach outperforms other abstraction algorithms [56], but this remains a concern.

'Work' in EMD is defined as a function of the ground distance between two elements, which is only well-defined if the proximity between elements also is. This is problematic for reasons outlined above. In addition, although those papers and this work focus on Hold'em variants, this measure ideally should be useful for any choice of variant as long as it shares basic mathematical properties. In Hold'em, we can try to place the available actions at any node on a spectrum - fold or check representing a bet of zero at the bottom, the maximum bet at the top - but other variants offer additional actions. In draw poker variants - popular with older players and in high-stakes mixed games - players have the option to discard cards and draw replacement cards. This action is a fundamental part of a strategy but cannot easily be modelled in this view of 'distance'.

In an earlier paper, Ganzfried uses a distance measure similar to EMD for opponent modelling by finding the closest strategy to a precomputed strategy given a limited set of observations of the opponent's play [58]. This approach is potentially relevant in the context of bot detection and its ability to operate efficiently in real-time is encouraging. This 'weight-shifting' (WS) algorithm works by comparing the frequency of an action in observations of an opponent with the frequency of that action in the baseline strategy and shifting the weights assigned in the baseline strategy until it is consistent with the real strategy; we can use the amount by which each weight or all weights are shifted as a distance measure. Experimental data showed that this approach outperformed weighted L_1 and L_2 measures for the purpose of constructing and responding to an opponent model.

The main concern with this measure in practice will be familiar from the discussion of opponent modelling: it does not extend well to larger variants. If a node in the game tree is visited rarely and we have few observations of the opponent's play at that point, our estimation of the opponent's 'real' frequency of any action at that node is necessarily unreliable. For the testing in [58] this was handled by waiting to compute and implement the opponent model until enough observations were gathered. Parts of the strategy that did not show up in the sample were assumed to be the same as in the baseline strategy. This is sensible for opponent modelling but does not help if we just want to evaluate the overall distance.

6.3 Testing

We saw empirical evidence in Section 4 that cross-set tabulation of play statistics are a quick and useful way to identify suspicious actors and that this is especially useful for players trying to confirm suspicions with limited resources or background knowledge. However, if we want to give a rigorous justification for this, we have to engage some difficult questions. This method is sensitive to the choice of statistics as adding or removing statistics from the input vector will change the total distance. The statistics given in HUDs and elsewhere are defined and chosen for their practical utility to players and are not guaranteed to be theoretically meaningful. Being informed when making this choice requires a sound understanding of what each statistic represents in terms of wider poker strategy and how much weight it deserves.

We must also be aware of how these statistics relate to each other. Some are empirically covariant: 'loose-aggressive' players earn this label from a common play pattern of playing more hands (higher VPIP) and playing them more aggressively (higher PFR, squeeze, CBet%). Some are theoretically covariant: a player who raises less often preflop (lower PFR) should make continuation bets on the flop more frequently as their range is stronger (higher Flop CBet%). Others are related by definition: Aggression Factor (AF) is calculated from the ratio of aggressive actions to calling actions and thus will vary with PFR. The concern here is that, if too many covariant statistics are used in the vector, a small change to one part of the player's strategy results in a disproportionately large change to the output distance.

The distance measure used in Section 4, highlighted in 4.5, can be expressed formally as:

$$d = (\sum_{i=1}^{k} w_i c_i (x_i - y_i)^n)^{1/n}$$

Where x and y are input vectors consisting of the chosen statistics and k is the number of statistics in the vector. Different values of n yield specific values of the Minkowski distance under the L_p norm: $n = 1$ gives the L_1 distance or 'Manhattan distance'; $n = 2$ gives the L_2 distance or Euclidean distance. For higher values of n the largest values of $(x_i - y_i)^n$ contribute more to the total distance so L_1 may prove to be better for this role since we care about small differences between statistics; this is especially true for high k. w_i represents the weight assigned to the statistic i, reflecting its relative importance. c_i represents our confidence in the statistic. This is usually based on how many relevant observations we have but can also reflect a belief that some observations are more reliable than others.

The Mahalanobis distance is a variant of L_2 distance that accounts for covariance between variables:

$$D_M = ((x - y)^T C^{-1} (x - y))^{1/2}$$

Where C is the covariance matrix. If C is the identity matrix, this is just the L_2 distance; if C is diagonal, we have the normalized L_2 distance. This works by decorrelating and standardizing the data and calculating the L_2 distance for that data. As above, we can assign weights to individual elements in the vector.

Its most prominent use in poker is in a paper by Yampolskiy and Govindaraju [59] in a comparison of distance measures for behavioural biometric methods. In this context it gave the worst performance of all measures tried (L_1, L_2, weighted L_2, Mahalanobis). However, the authors attribute this to possible problems caused by the normalization procedure which should not apply to the data used below. In the absence of factors like this, Mahalanobis distance appears to be the most theoretically sound for this task; the pre-testing hypothesis is that this will give the best performance.

6.3.1 Methodology

Hand histories were collected for a population containing suspected bots in a specific No Limit Hold'em game on a popular poker site. A random sample of these hands was selected to serve as the smallest data set. Groups of hands were randomly added to this until the total for the next sample size was reached. The commercial Hold'em Manager software was used in conjunction with the PokerParser tool developed for this project and described in Section 7 to derive the desired statistics. The suspected bots are joined by a composite 'player' representing the average bot. After standardizing the data, these were compared against the average for the whole sample as well as several ‘baseline’ players; these were chosen from the high-volume regulars in the game to make the effective sample size higher, but this should not skew the results unless high-volume players are unrepresentative of the target population. Since the bots themselves and the players most interested in these findings are high-volume, this seemed appropriate.

The scope of this experiment depends on three factors:

Sample construction

To be effective in real time security measures must work with a small sample size. For this experiment there was a trade-off between accurately capturing the problems raised by this and ensuring the results of the experiment was reliable and significant. This sample also had to be large enough to contain most of the suspected bots. A set of 5,000 hands was used as it featured enough of these accounts and it is easy for a player to play this many hands during a session. The other sample sizes were: 15,000; 50,000; 115,000; 175,000. This does not cover a full range of sizes - investigations of long-term bot activity can cover hundreds of thousands or millions of hands - but it was hoped that these would be enough to demonstrate if and how the utility of distance measures varies with the sample size. Note that this figure represents the total number of hands in the sample: the number of hands played against any one opponent will be much lower.

Statistics

A set of important statistics was identified by analyzing previous investigations and consulting professional players:

VPIP: How often the player voluntarily puts money into the pot pre-flop
PFR: How often the player raises the blinds before the flop
3Bet PF: How often the player re-raises the blinds pre-flop
Squeeze: How often the player 3-bets when someone has called the initial raise
Fold to 3Bet PF%: How often the player folds to a 3-bet pre-flop after raising
Postflop Agg%: How often the player takes aggressive actions like betting or raising
[Street] CBet%: Specifically, how often the player makes a continuation bet on [street]
Fold to [Street] CBet%: How often the player folds to a continuation bet on [street]
WTSD%: How often the player went to showdown after seeing the flop
W$WSF%: How often the player won part of the pot when they saw the flop

Collectively, these give a reliable summation of a player's tendencies and test a distance metric's ability to cope with covariance.

Measures

Five measures were chosen: L1, weighted L1, L2, weighted L2, and Mahalanobis. It is natural to compare distance measures in the same norm and bothL1and L2are commonly referenced in poker research. We also want to see how a simple weighting will affect the results and how this depends on the choice of measure. This weighting is applied after the data is standardized and consists of these multipliers:

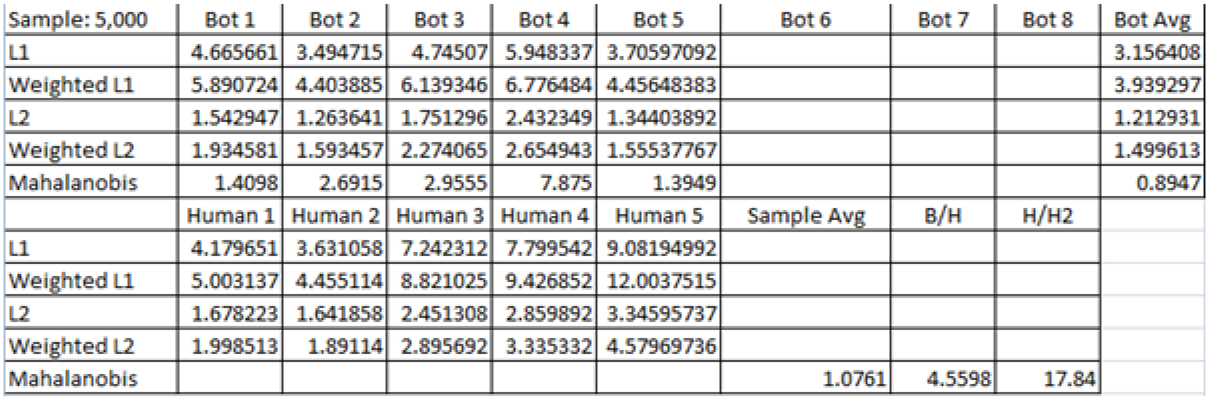
2: W$WSF
1.5: VP$IP, PFR, Postflop Agg%, WTSD%
1: 3Bet, W$SD, Flop CBet%, Fold to Flop CBet, Squeeze

The 1.5 multiplier was applied to statistics mentioned frequently as sources of unusual behaviour. Extra weight was given to W$WSF as this was identified as a distinguishing feature for this group of suspected bots.

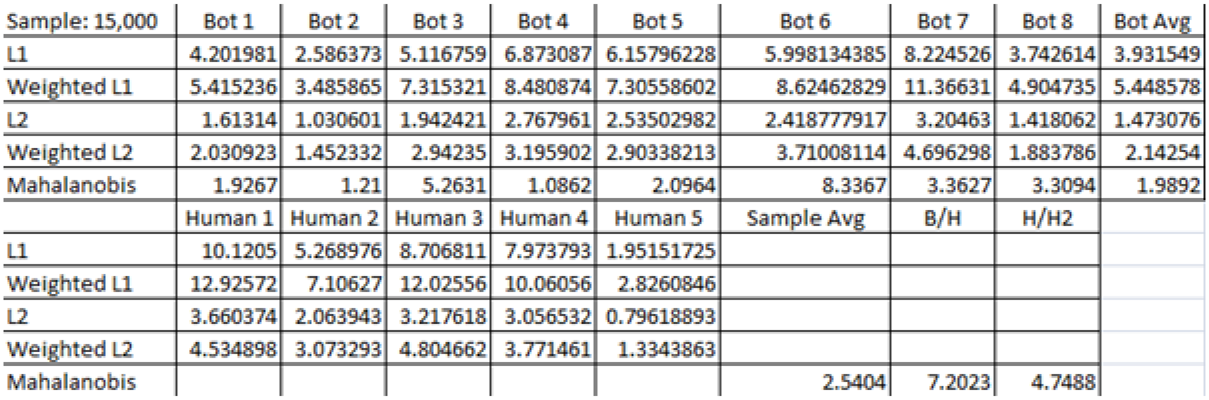
We observed that the Mahalanobis distance has appealing properties and wanted to see how it performs. The high-volume baseline players were used as a group to compare with individual suspect accounts. Additionally, a second group of baseline players was compared against the first to test if the overall difference between bots and humans was greater than for two sets of humans. The Mahalanobis distance is typically used to compare two distributions rather than a single vector and a distribution, so this final step ensures that this is featured in the experiment.

This range of measures only represents the approach advocated in this section. It would be useful to include other approaches but this often presents considerable logistical challenges: EMD is an obvious choice given its previous use in poker research but it is impractical to evaluate for a data set this large. We could create a small abstraction, map a player's observed strategy to actions in this abstraction, and then compute the EMD, but the result would be largely determined by errors introduced in the abstraction. By restricting the analysis to similar measures, we are left with an easier, self-contained problem. Nevertheless, finding a way to compare different 'types' of distance measure in this context is a priority for future work.

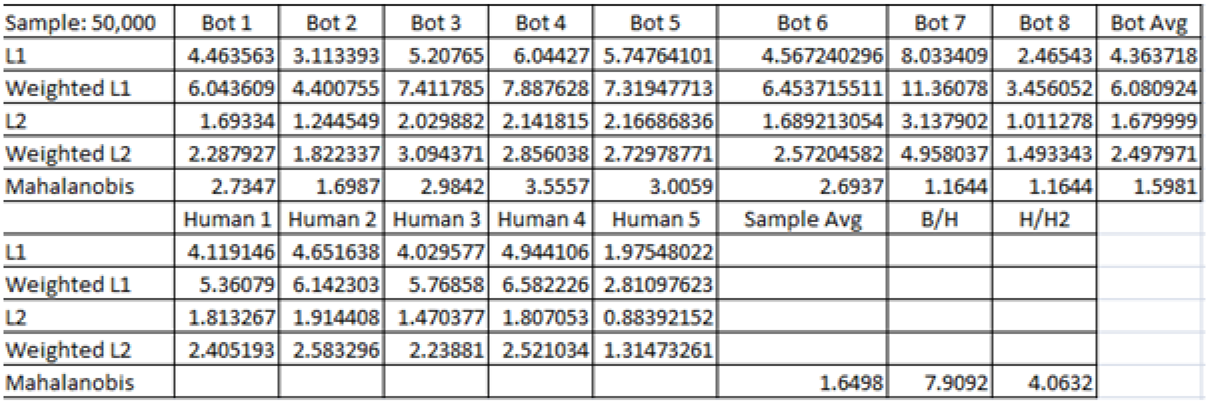
6.3.2 Results



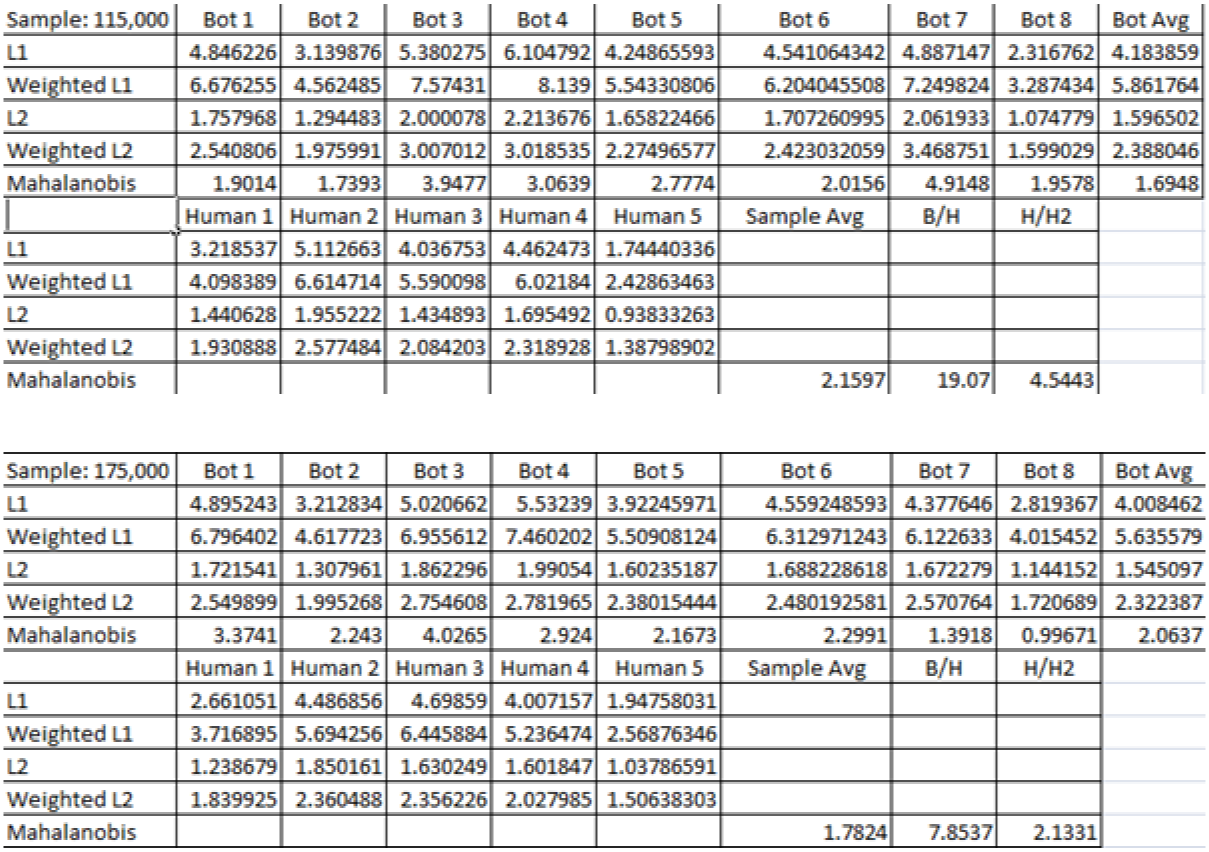
Sample: 5,000	Bot 1	Bot 2	Bot 3	Bot 4	Bot 5	Bot 6	Bot 7	Bot 8	Bot Avg
L1	4.665661	3.494715	4.74507	5.948337	3.70597092				3.156408
Weighted L1	5.890724	4.403885	6.139346	6.776484	4.45648383				3.939297
L2	1.542947	1.263641	1.751296	2.432349	1.34403892				1.212931
Weighted L2	1.934581	1.593457	2.274065	2.654943	1.55537767				1.499613
Mahalanobis	1.4098	2.6915	2.9555	7.875	1.3949				0.8947
	Human 1	Human 2	Human 3	Human 4	Human 5	Sample Avg	B/H	H/H2	
L1	4.179651	3.631058	7.242312	7.799542	9.08194992				
Weighted L1	5.003137	4.455114	8.821025	9.426852	12.0037515				
L2	1.678223	1.641858	2.451308	2.859892	3.34595737				
Weighted L2	1.998513	1.89114	2.895692	3.335332	4.57969736				
Mahalanobis						1.0761	4.5598	17.84	



Sample: 15,000	Bot 1	Bot 2	Bot 3	Bot 4	Bot 5	Bot 6	Bot 7	Bot 8	Bot Avg
L1	4.201981	2.586373	5.116759	6.873087	6.15796228	5.998134385	8.224526	3.742614	3.931549
Weighted L1	5.415236	3.485865	7.315321	8.480874	7.30558602	8.62462829	11.36631	4.904735	5.448578
L2	1.61314	1.030601	1.942421	2.767961	2.53502982	2.418777917	3.20463	1.418062	1.473076
Weighted L2	2.030923	1.452332	2.94235	3.195902	2.90338213	3.71008114	4.696298	1.883786	2.14254
Mahalanobis	1.9267	1.21	5.2631	1.0862	2.0964	8.3367	3.3627	3.3094	1.9892
	Human 1	Human 2	Human 3	Human 4	Human 5	Sample Avg	B/H	H/H2	
L1	10.1205	5.268976	8.706811	7.973793	1.95151725				
Weighted L1	12.92572	7.10627	12.02556	10.06056	2.8260846				
L2	3.660374	2.063943	3.217618	3.056532	0.79618893				
Weighted L2	4.534898	3.073293	4.804662	3.771461	1.3343863				
Mahalanobis						2.5404	7.2023	4.7488	



Sample: 50,000	Bot 1	Bot 2	Bot 3	Bot 4	Bot 5	Bot 6	Bot 7	Bot 8	Bot Avg
L1	4.463563	3.113393	5.20765	6.04427	5.74764101	4.567240296	8.033409	2.46543	4.363718
Weighted L1	6.043609	4.400755	7.411785	7.887628	7.31947713	6.453715511	11.36078	3.456052	6.080924
L2	1.69334	1.244549	2.029882	2.141815	2.16686836	1.689213054	3.137902	1.011278	1.679999
Weighted L2	2.287927	1.822337	3.094371	2.856038	2.72978771	2.57204582	4.958037	1.493343	2.497971
Mahalanobis	2.7347	1.6987	2.9842	3.5557	3.0059	2.6937	1.1644	1.1644	1.5981
	Human 1	Human 2	Human 3	Human 4	Human 5	Sample Avg	B/H	H/H2	
L1	4.119146	4.651638	4.029577	4.944106	1.97548022				
Weighted L1	5.36079	6.142303	5.76858	6.582226	2.81097623				
L2	1.813267	1.914408	1.470377	1.807053	0.88392152				
Weighted L2	2.405193	2.583296	2.23881	2.521034	1.31473261				
Mahalanobis						1.6498	7.9092	4.0632	



Sample: 115,000	Bot 1	Bot 2	Bot 3	Bot 4	Bot 5	Bot 6	Bot 7	Bot 8	Bot Avg
L1	4.846226	3.139876	5.380275	6.104792	4.24865593	4.541064342	4.887147	2.316762	4.183859
Weighted L1	6.676255	4.562485	7.57431	8.139	5.54330806	6.204045508	7.249824	3.287434	5.861764
L2	1.757968	1.294483	2.000078	2.213676	1.65822466	1.707260995	2.061933	1.074779	1.596502
Weighted L2	2.540806	1.975991	3.007012	3.018535	2.27496577	2.423032059	3.468751	1.599029	2.388046
Mahalanobis	1.9014	1.7393	3.9477	3.0639	2.7774	2.0156	4.9148	1.9578	1.6948
	Human 1	Human 2	Human 3	Human 4	Human 5	Sample Avg	B/H	H/H2	
L1	3.218537	5.112663	4.036753	4.462473	1.74440336				
Weighted L1	4.098389	6.614714	5.590098	6.02184	2.42863463				
L2	1.440628	1.955222	1.434893	1.695492	0.93833263				
Weighted L2	1.930888	2.577484	2.084203	2.318928	1.38798902				
Mahalanobis						2.1597	19.07	4.5443	

Sample: 175,000	Bot 1	Bot 2	Bot 3	Bot 4	Bot 5	Bot 6	Bot 7	Bot 8	Bot Avg
L1	4.895243	3.212834	5.020662	5.53239	3.92245971	4.559248593	4.377646	2.819367	4.008462
Weighted L1	6.796402	4.617723	6.955612	7.460202	5.50908124	6.312971243	6.122633	4.015452	5.635579
L2	1.721541	1.307961	1.862296	1.99054	1.60235187	1.688228618	1.672279	1.144152	1.545097
Weighted L2	2.549899	1.995268	2.754608	2.781965	2.38015444	2.480192581	2.570764	1.720689	2.322387
Mahalanobis	3.3741	2.243	4.0265	2.924	2.1673	2.2991	1.3918	0.99671	2.0637
	Human 1	Human 2	Human 3	Human 4	Human 5	Sample Avg	B/H	H/H2	
L1	2.661051	4.486856	4.69859	4.007157	1.94758031				
Weighted L1	3.716895	5.694256	6.445884	5.236474	2.56876346				
L2	1.238679	1.850161	1.630249	1.601847	1.03786591				
Weighted L2	1.839925	2.360488	2.356226	2.027985	1.50638303				
Mahalanobis						1.7824	7.8537	2.1331	

Figure 8: Results of the test; B/H is the distance between the bot group and the human group, H/H2 between the two human groups

6.3.3 Evaluation

We can see from Figure 7 that this test was largely inconclusive. Predictably, the results were highly erratic for the smallest sample and improved with the size of the sample. Early samples displayed a high variance between distance measures, especially the Mahalanobis distance. For the largest sample, there was no clear pattern that distinguished bots from humans. An obvious reason is that the largest sample was not that large for individual players. In the 175,000 hand sample, the most active account in the entire set was Bot 1 who amassed around 25,000 hands and only six played more than 10,000; the average account played fewer than 500 hands. This meant that the issues with small samples continued to apply even as the sample grew to a seemingly respectable size. Requiring a large enough data set to avoid this problem adds to the difficulty in finding a suitable data set and restricts these investigations to people who have access to this data; it also makes the test less applicable to smaller, more common cases of bot use.

The choice of hands merits some discussion. The samples were constructed by taking the smallest sample, retaining most of the hands, and adding new hands until the next threshold was reached. The alternative is to select hands randomly from the data set for each sample (perhaps even adjusting so that the same hands rarely or never feature in multiple samples). We chose to build on previous samples because this preserves the continuity of earlier results and isolates the effect of adding new hands to the sample. However, this ensures that any irregularities in early samples will feature in all samples. It also compounds the sample size problem: if a player appears less often than expected in the additional hands, the difference from their profile in the previous sample is minimal. Random selection faces the opposite problem: if a player is underrepresented in the new sample, or their behaviour in the new sample does not accurately reflect their tendencies, this will skew the results. A more comprehensive study could test both approaches for varied sample sizes and contents.

We also chose not to treat hands with multiple suspects differently. Implicit collusion is a natural occurrence in multiplayer games and members of a bot ring are likely to collude explicitly. This change to the incentives facing a bot when seated with another member will manifest in the statistics and therefore in the eventual distance. This choice was made because quantifying this effect is difficult and modelling it improperly seemed worse than not modelling it at all. Even though this test used a list of suspects, it was intended to reflect the state of knowledge of someone starting an investigation with no names of suspects and only a vague sense that something is amiss. However, these results suggest to some extent that the approach used here is only suitable for a directed investigation on an extensive data set. In this case, assuming this knowledge and distinguishing between hands like this (or based on other properties) will yield better results. In general, expert knowledge or external information can be used to choose criteria to define the sample.

The use of weighting did not appear to make either L_1 or L_2 distance more effective. The choice of multiplier was somewhat arbitrary; we know that W$WSF% is more important but it is unclear how to quantify this. We may have to seek out findings from distance weighting tests in other domains to inform our choices here.

An interesting finding is that, even when suspects' individual Mahalanobis distances from the baseline human sample were unremarkable compared to its distance from the mean, the group differences between bots and humans were stark. In Figure 8, the massive difference between B/H and H/H2 for the 5,000 hand sample can be attributed to sample size, but the even larger difference in the other direction for the 115,000 hand sample stands out and has no obvious explanation. It is hard to reconcile this with the clear pattern in the other observations, which would otherwise be a promising way to detect groups of bots.

Some problems can be attributed to the data set. Large databases of hand histories can be obtained with some effort but it is hard to find a population that contains known bots; this was the best data set that could be found under these conditions. Its suitability was due in part to its bot problem not being obvious: the suspects are not uniform in their play and for the most part resemble human players. It is possible that some of the measures used here actually performed well under the circumstances. This highlights the lack of a well-defined success criterion as a flaw in our methodology (for instance, we could set a threshold that 'designates' an account as a bot and cross-reference accounts that pass this threshold with the list of suspects but then we need to know how to judge where this threshold is). Without this, evaluation becomes much harder. For instance, the unnormalized Euclidean distance measure used in Section 5.4.5 certainly does better at illustrating the difference between bots and humans, and in that investigation it was obviously successful, but the lack of normalization is a severe theoretical flaw.

In summary, the results of this experiment do not make a strong case for or against any distance measure. Contrary to the hypothesis, Mahalanobis distance did not outperform other measures but it displayed interesting properties that deserve further study. Experiments like this can be a productive way of finding deviant players in a population as long as care is taken to address the problems displayed above.

7. PokerParser

For the implementation aspect of the project, a basic Java program was developed for use alongside commercial analysis software such as PokerTracker or Hold'em Manager. These programs allow the generation and comparison of player statistics but are limited in their ability to directly compare these; users would have to perform more complicated comparisons manually (such as by downloading the database and using SQL). PokerParser is a user-friendly tool for these comparisons.

The user chooses the .csv file (typically a report exported from the software above) and how many comparisons they want to make via the console, then gives details for each: the name of the statistic, the basis for comparison (a number or name of a player), the negative range, and the positive range. The Univocity parser library is used to scan the file before analysis is done.

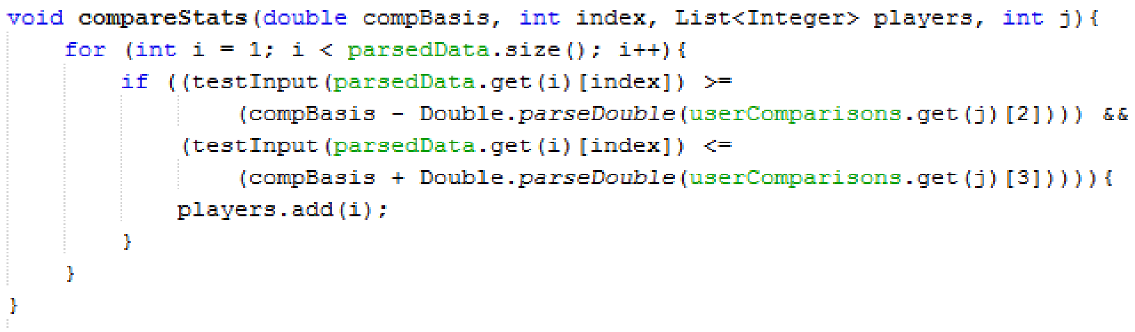
Although this was developed for use in poker, this tool can easily be applied to other data sets e.g. searching for possible discrepancies in accounting data.

A limited version with a GUI was developed for demonstration purposes (see Figure 10).

7.1 Evaluation

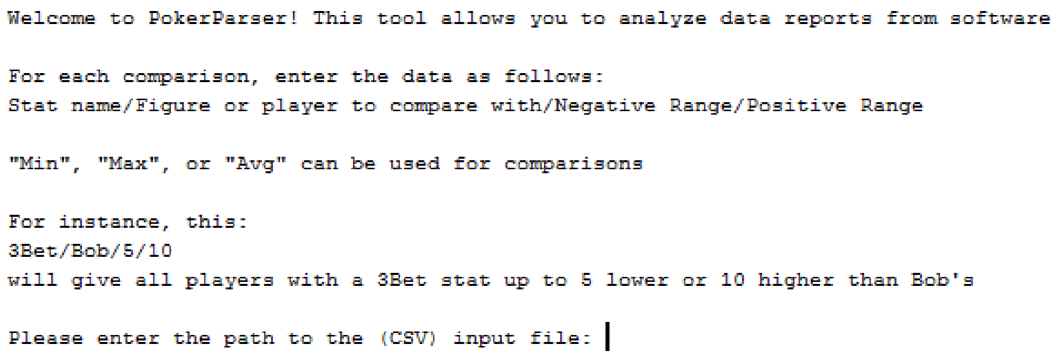
As the focus of this thesis is the theoretical and investigatory work, this is submitted more as a proof of concept than as a finished product. It does the job but could stand to have more depth. Because of its simplicity the program is structured as a single class; adding more complicated functionality would require a proper object-oriented approach. Basic functions that could be added include using previous queries to enable queries where the basis for comparison is not known yet (e.g. 'Find the player that fits Comparison 1 and Comparison 2, then use them as the basis for Comparison 3').

User trials with poker players from a range of experience levels were planned but the intended users were unable to commit time over the holiday season. The current interface is a little clunky and unintuitive, which is a result of my using Swing instead of a more sophisticated graphics library. The console version allows for more flexibility in comparisons, so it is likely they would have preferred it over the GUI anyway.



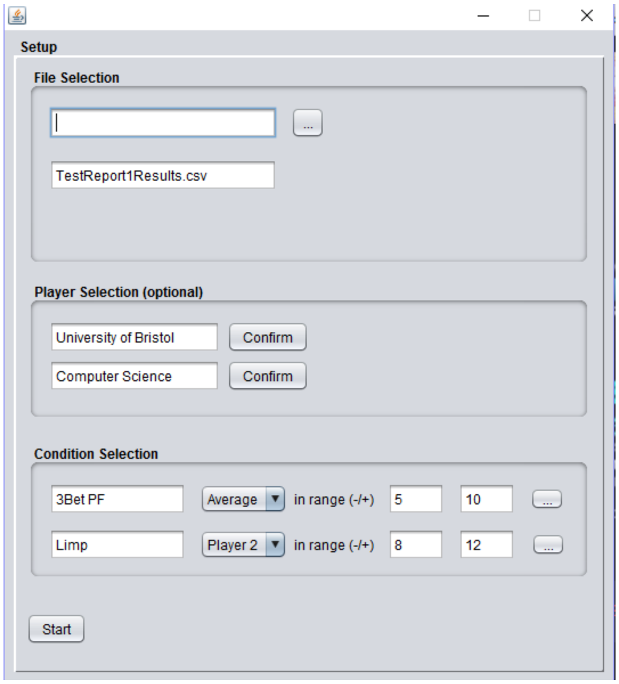
```
void compareStats(double compBasis, int index, List<Integer> players, int j){
    for (int i = 1; i < parsedData.size(); i++){
        if ((testInput(parsedData.get(i)[index]) >=
                (compBasis - Double.parseDouble(userComparisons.get(j)[2]))) &&
            (testInput(parsedData.get(i)[index]) <=
                (compBasis + Double.parseDouble(userComparisons.get(j)[3])))){
            players.add(i);
        }
    }
}
```

Figure 9: The method used to find the players who meet the given criteria



```
Welcome to PokerParser! This tool allows you to analyze data reports from software

For each comparison, enter the data as follows:
Stat name/Figure or player to compare with/Negative Range/Positive Range

"Min", "Max", or "Avg" can be used for comparisons

For instance, this:
3Bet/Bob/5/10
will give all players with a 3Bet stat up to 5 lower or 10 higher than Bob's

Please enter the path to the (CSV) input file:
```

Figure 10: PokerParser run via console



TestReport1Results.csv

University of Bristol

Computer Science

3Bet PF | Average | in range (-/+) | 5 | 10

Limp | Player 2 | in range (-/+) | 8 | 12

Figure 11: The demo GUI for PokerParser

8. Summary, evaluation, and future work

Evaluation for specific sections is given in Sections 3.3.1, 6.3.3, 7.1.

Overall, the added value of this project largely consists of its exploration of new territory for poker research. For topics like opponent modelling that are already covered extensively in the literature, this analysis moved forward in a new direction. For other topics, such as human understanding of strategic concepts and its relevance for poker pedagogy, this was one of the first formal treatments. The downside of this is that, as an amateur computer scientist (and amateur poker hobbyist) with a timeline of a few months, the scope of this project had to be realistic. Diving too far into one topic, or being stretched too thin between many, would harm the overall project. Lacking a frame of reference for the new material meant that I could not be sure that all of it was theoretically sound or that some of the ideas about human learning etc. would be borne out in practice; developing and testing these ideas is the main area of future work. I hope and believe that I have struck a good balance between introducing and generating new insights and building on existing material.

Theory

Before the focus of the project had been chosen or any details decided, the original goal was to give a general account of how humans and machines coexist in online poker. Section 3 gave an extensive abstract account of man-machine interaction in strategic games and reinforced this with insights from other domains. Testing these theories was outside the scope of this project but is needed to put these on a more solid foundation.

This section ended with suggestions for modifications to popular poker variants that would be enjoyable and novel for humans while frustrating bots' attempts to play. Again, these are mere suggestions until they are tested, but it is impossible to test these on a large enough scale without the co-operation of poker sites. The basic idea is very promising and open-ended so there may well be modifications more effective than those given here.

Future work here involves testing with educationalists, poker coaches and theorists, and poker students to develop a well-rounded account of poker pedagogy that will hopefully be useful in both behavioural game theory and education. Similar surveys to the one carried out in Section 3 should be devised for poker coaches and their students as well as poker authors and consumers of instructional material.

The concept of freestyle chess could be extended to other games to test how human thought processes and machines can complement each other. One idea is to invent a poker variant unusual enough that no specific poker knowledge will carry over (only the general 'game sense' that constitutes talent or which a player develops over time) and allow players to use powerful machine aids to assist their play in real-time against each other; the most successful players should be those who can use the machines to their advantage.

In general, it is worth exploring the use of bots in educational contexts; we could deliberately introduce imbalances into a bot's play and challenge a student to identify these within certain constraints (e.g. to make the problem harder, this has to be done over a limited number of hands). The prevalence of HUDs, PioSOLVER etc. suggest that dedicated players are willing and able to learn valuable lessons from analysis offered by machines. It may be instructive to compare poker players' understanding of game theory or AI with that of other occupations, especially those that do not explicitly require mathematical knowledge.

On the opponent modelling front, it is becoming increasingly clear that some of the old assumptions about the inefficiency of neural networks no longer apply. This has a lot of promise for fraud detection, as an artificially intelligent system is well suited at classifying these behaviour patterns. In addition, the idea of a team of agents looks increasingly salient given the emergence of bot rings. From the arguments advanced here, this is a recommended direction for future research in opponent modelling.

Research

A selection of bot investigations were used to demonstrate the efficacy of the statistical cross-set approach. However, this revealed little about bot investigations as a whole. These were chosen because they were successful but there are many more bot investigations that never get off the ground because of poor methodology, lack of evidence, or simply loss of interest. A more comprehensive study that also looks at these would help future investigators have confidence in their methods and findings.

A detailed survey was conducted to give the first public account of the perception of bot use in the poker community. Evaluation of this is found in Section 3.3.1. This section also addressed popular misconceptions within the community about the limits of AI and its applicability to poker. It would have been useful to link these tasks from the start so that the survey could give a better sense of how well these concepts are understood. It is also worth tracking if and how this understanding has evolved over time (say, with a study of forum posts on game theory or AI over time).

Implementation

See 7.1 for a full explanation. Thinking in terms of the work's other themes, future work could monitor how a program like this is used; what types of query do users often make? This would allow the tool itself to be refined as well as giving some insight into what areas of poker strategy are most interesting or relevant to players.

A note on sources

Part of the motivation for this project was the lack of existing research into important topics in poker. As such, these discussions relied on a range of non-traditional sources including forum posts and interviews. To ensure rigour, these were only referenced if I could find similar arguments from other reputable sources.

Private correspondence with several professional and independent researchers was helpful in preparing this paper and is referenced as appropriate. Some consisted of just a follow-up question about some public statement; others were more lengthy conversations. In particular I am grateful to Sam Ganzfried, Michael Johanson, and Eric Jackson (developer of the Slumbot program that finished 2nd in the Annual Computer Poker Competition) for their detailed feedback.

Many of the papers cited here form part of larger works, specifically Sam Ganzfried and Michael Johanson's PhD theses. Rather than citing these works as a whole, invididual papers are cited for the sake of clarity and to give due credit to the co-authors. The theses are listed in the Bibliography as useful reference material.

References

Web pages accessed 12 January 2017.

[1] J. Schaeffer, "A Gamut of Games", *AI Magazine*, Vol. 22, no. 3, 2001.

[2] V. Lisý et al., "Counterfactual Regret Minimization in Sequential Security Games", *Computer Poker Research Group*, 2016.

[3] M. Bowling and M. Johanson, audio interview for *Thinking Poker*, Episodes 79 & 110, 2014/2015.

[4] D. Billings, "Computer Poker", *Computer Poker Research Group*, 1995.

[5] J. Hawkin, "Automated Abstraction of Large Action Spaces in Imperfect Information Extensive-Form Games", *Computer Poker Research Group,* 2014.

[6] M. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[7] October 2011 High Stakes discussion, forum posting, http://forumserver.twoplustwo.com/19/high-stakes-pl-nl/october-official-hsnl-ftp-one-time-1106746/index2.html

[8] Annual Computer Poker Competition, http://www.computerpokercompetition.org/

[9] H. Kuhn, "Simplified Two-Person Poker", *Contributions to the Theory of Games*, Princeton University Press, 1950.

[10] B. Chan and J. Ankenman, *The Mathematics of Poker*, Two Plus Two Publishing, 2006.

[11] M. Johanson, "Measuring the Size of Large No-Limit Poker Games", *Computer Poker Research Group*, 2013.

[12] S. Ganzfried and F. Yusuf, "Learning Human-Understandable Strategies", *AAAI Conference on Artificial Intelligence*, 2017.

[13] K. Waugh et al., "Abstraction Pathologies in Extensive Games", *Computer Poker Research Group*, 2009.

[14] J. Levermann, audio interview for *Thinking Poker*, Episode 54, 2013.

[15] Tournament Poker Edge, https://www.tournamentpokeredge.com

[16] Thinking Poker, http://www.thinkingpoker.net/coaching/

[17] M. Laakasuo et al., "Emotional and Social Factors Influence Poker Decision Making Accuracy", *Journal of Gambling Studies*, 2014.

[18] M. Craig, *The Professor, the Banker, and the Suicide King: Inside the Richest Poker Game of All Time*, Grand Central Publishing, 2005.

[19] PokerStars live traffic report, *PokerScout*, http://www.pokerscout.com/SiteDetail.aspx?site=PokerStars&ab=410645248

[20] Personal correspondence, Noah Stephens-Davidowitz, www.noahsd.com and www.nsdpoker.com

[21] S. Ganzfried, "Optimal Number of Choices in Rating Contexts", *Neural Information Processing Systems*, 2016.

[22] S. Ganzfried and T. Sandholm, "Safe Opponent Exploitation", *ACM Transactions on Economics and Computation*, 2015.

[23] J. Preston and M. Bishop (eds.), *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, Oxford University Press, 2002.

[24] Chess News, "Amateurs beat GMs in PAL/CSS Freestyle", http://en.chessbase.com/post/pal-c-report-from-the-dark-horse-s-mouth, 2005.

[25] D. Billings et al., "The challenge of poker", *Artificial Intelligence* 134, 2002.

[26] N. Bard, "Online Agent Modelling in Human-Scale Problems", *Computer Poker Research Group*, 2016.

[27] D. Billings et al., "Improved Opponent Modelling in Poker", *Computer Poker Research Group*, 2000.

[28] B. Hoehn et al., "Effective Short-Term Opponent Exploitation in Simplified Poker", *Computer Poker Research Group*, 2005.

[29] M. Johanson, "Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player", *Computer Poker Research Group*, 2007.

[30] M. Johanson and M. Bowling, "Data Biased Robust Counter Strategies", *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.

[31] J. Rubin and I. Watson, "Computer poker: A review", *Artificial Intelligence* 175, 2011.

[32] A. Davidson, "Using Artificial Neural Networks to Model Opponents in Texas Hold'em", submitted to *University of Alberta*, 1999.

[33] A. Davidson, "Opponent Modelling in Poker: Learning and Acting in a Hostile Environment", *Computer Poker Research Group*, 2002.

[34] N. Yakovenko, audio interview for *Thinking Poker*, Episode 155, 2016.

[35] Personal correspondence, Nikolai Yakovenko

[36] Personal correspondence, Matt Mazur, https://mattmazur.com/

[37] Maitrepierre et al., "Adaptive play in Texas Hold'em poker", *European Conference on Artificial Intelligence*, 2008.

[38] N. Stephens-Davidowitz and T. Bakker, "Independent Cake Poker Investigation", http://www.nsdpoker.com/wp-content/uploads/2011/11/Cake_Investigation_Results_11_27_2010.pdf, 2010.

[39] Personal correspondence, Michael Johanson

[40] A. Gilpin et al., "Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker", *Association for the Advancement of Artificial Intelligence*, 2007.

[41] F. Southey et al., "Bayes' Bluff: Opponent Modelling in Poker", *Computer Poker Research Group*, 2005.

[42] 888poker, "Playing Against Poker Robots", https://www.888poker.com/magazine/strategy/playing-against-poker-bots/

[43] N. Stephens-Davidowitz, "Proof that Nick "Stoxtrader" Grudzien, "Littlezen", and "Kinetica" Colluded", www.nsdpoker.com/wp-content/uploads/2011/12/stoxtrader.pdf, 2010.

[44] http://forumserver.twoplustwo.com/28/internet-poker/bots-party-insiders-view-1474198/

[45] http://forumserver.twoplustwo.com/showpost.php?p=50228755&postcount=1

[46] http://forumserver.twoplustwo.com/showpost.php?p=47169453&postcount=1

[47] Personal correspondence, Casper Greve

[48] M. Mazur, "Improving PokerStars's Bot Detection Techniques" et al., https://mattmazur.com/category/poker-bot/, 2009.

[49] UK Gambling Commission consultation, "Remote gambling and software technical standards", October 2016.

[50] P. Mazrooei et al., "Automatic Collusion Detection in Sequential Games", *Computer Poker Research Group*, 2013.

[51] S. Rachev , *Probability Metrics and the Stability of Stochastic Models*, John Wiley & Sons, 1991.

[52] Personal correspondence, Sam Ganzfried

[53] S. Ganzfried, "Reflections on the First Man vs. Machine No Limit Texas Hold'Em Competition", presented at *2016 World Congress of the Game Theory Society*, 2015.

[54] M. Bowling et al., "Heads-up Limit Hold'em Poker is Solved", *Computer Poker Research Group*, 2015.

[55] E. Jackson, written interview, *PartTimePoker*, http://www.parttimepoker.com/poker-ai-with-eric-jackson-part-3

[56] S. Ganzfried and T. Sandholm, "Potential-aware Imperfect-recall Abstraction with Earth Mover's Distance in Imperfect-Information Games", *Association for the Advancement of Artificial Intelligence*, 2014

[57] M. Johanson et al, "Evaluating State-Space Abstractions in Extensive-Form Games", *Computer Poker Research Group*, 2013.

[58] S. Ganzfried and T. Sandholm, "Game Theory-Based Opponent Modelling in Large Imperfect-Information Games", Proceedings of the *International Conference on Autonomous Agents and Multiagent Systems*, 2011.

[59] R. Yampolskiy and V. Govindaraju, "Similarity Measure Functions for Strategy-Based Biometrics", *International Journal of Computer, Electrical, Automation, Control, and Information Engineering*, 2008.

Bibliography

S. Ganzfried, "Computing Strong Game-Theoretic Strategies and Exploiting Suboptimal Opponents in Large Games", submitted as PhD dissertation to *Carnegie Mellon University*, 2015.

M. Johanson, "Robust Strategies and Counter-Strategies: From Superhuman to Optimal Play", submitted as PhD dissertation to *Computer Poker Research Group*, 2016.

M. Moravcik, "DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker", *arXiv*, 2017.

T. Sandholm, "The State of Solving Large Incomplete-Information Games, and Application to Poker", *Association for the Advancement of Artificial Intelligence*, 2010.

O. Tammelin et al., "Solving Limit Texas Hold'em", Proceedings of the *24th International Joint Conference on Artificial Intelligence*, 2015.

R. Gibson, "Regret Minimization in Games and the Development of Champion Multiplayer Computer Poker-Playing Agents", submitted as PhD dissertation to *Computer Poker Research Group*, 2014.